



Calhoun: The NPS Institutional Archive
DSpace Repository

Theses and Dissertations

Thesis and Dissertation Collection

1976-09

Time dependent holographic interferometry
and finite-element analysis of heat transfer
within a rectangular enclosure

Braun, Gerald Paul

Monterey, California. Naval Postgraduate School

<http://hdl.handle.net/10945/17980>

Downloaded from NPS Archive: Calhoun



Calhoun is a project of the Dudley Knox Library at NPS, furthering the precepts and goals of open government and government transparency. All information contained herein has been approved for release by the NPS Public Affairs Officer.

Dudley Knox Library / Naval Postgraduate School
411 Dyer Road / 1 University Circle
Monterey, California USA 93943

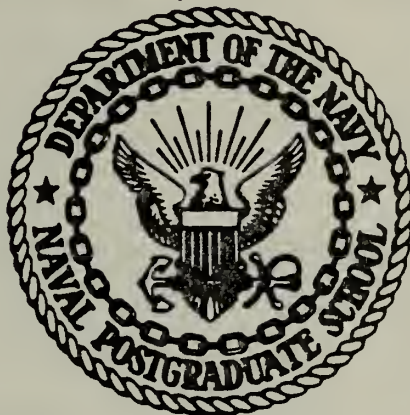
<http://www.nps.edu/library>

TIME DEPENDENT HOLOGRAPHIC INTERFEROMETRY
AND FINITE-ELEMENT ANALYSIS
OF HEAT TRANSFER WITHIN A
RECTANGULAR ENCLOSURE

Gerald Paul Braun

NAVAL POSTGRADUATE SCHOOL

Monterey, California



THESIS

TIME DEPENDENT HOLOGRAPHIC INTERFEROMETRY
AND FINITE-ELEMENT ANALYSIS
OF HEAT TRANSFER WITHIN A
RECTANGULAR ENCLOSURE

by

Gerald Paul Braun

September 1976

Thesis Advisor:

D. J. Collins

Approved for public release; distribution unlimited.

T174963

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS
BEFORE COMPLETING FORM

| | | | |
|---|--|-----------------------|---|
| 1. REPORT NUMBER | | 2. GOVT ACCESSION NO. | 3. RECIPIENT'S CATALOG NUMBER |
| 4. TITLE (and Subtitle) Time Dependent Holographic Interferometry and Finite-Element Analysis of Heat Transfer within a Rectangular Enclosure | | | 5. TYPE OF REPORT & PERIOD COVERED Master's Thesis September 1976 |
| 7. AUTHOR(s) Gerald Paul Braun | | | 6. PERFORMING ORG. REPORT NUMBER |
| 9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Postgraduate School Monterey, California 93940 | | | 8. CONTRACT OR GRANT NUMBER(s) |
| 11. CONTROLLING OFFICE NAME AND ADDRESS Naval Air Systems Command Department of the Navy Washington, D. C. 20360 | | | 10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS |
| 14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Naval Postgraduate School Monterey, California 93940 | | | 12. REPORT DATE September 1976 |
| | | | 13. NUMBER OF PAGES 195 |
| | | | 15. SECURITY CLASS. (of this report) Unclassified |
| | | | 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE |
| 16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited. | | | |
| 17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report) | | | |
| 18. SUPPLEMENTARY NOTES | | | |
| 19. KEY WORDS (Continue on reverse side if necessary and identify by block number) Holography Finite-Element Convective Heat Transfer | | | |
| 20. ABSTRACT (Continue on reverse side if necessary and identify by block number) In this thesis, the finite-element method was developed to numerically analyze heat transfer by laminar natural convection within a rectangular cavity, a classical fluid flow problem. A second auxiliary case study involving Couette flow was included to test the flexibility of this analysis technique. Analyzing heat flows experimentally was also explored utilizing holographic interferometry. Specific problems | | | |

20. Abstract (cont'd)

encountered during this phase of research are presented with appropriate comments.

Time Dependent Holographic Interferometry and
Finite-Element Analysis of Heat Transfer
within a Rectangular Enclosure

by

Gerald Paul Braun
Lieutenant, United States Navy
B.S.E.E., The University of Toledo, 1968

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN AERONAUTICAL ENGINEERING

from the
NAVAL POSTGRADUATE SCHOOL
September 1976

ABSTRACT

In this thesis, the finite-element method was developed to numerically analyze heat transfer by laminar natural convection within a rectangular cavity, a classical fluid flow problem. A second auxiliary case study involving Couette flow was included to test the flexibility of this analysis technique.

Analyzing heat flows experimentally was also explored utilizing holographic interferometry. Specific problems encountered during this phase of research are presented with appropriate comments.

TABLE OF CONTENTS

| | |
|--|----|
| NOMENCLATURE ----- | 7 |
| I. INTRODUCTION ----- | 9 |
| A. HOLOGRAPHIC INTERFEROMETRY ----- | 9 |
| B. CONCEPT AND HISTORY OF THE FINITE- ELEMENT METHOD ----- | 12 |
| II. FUNDAMENTAL THEORY OF FINITE-ELEMENT METHOD --- | 17 |
| A. VARIATIONAL PRINCIPLE ----- | 23 |
| B. FINITE-ELEMENT METHOD AND THE RITZ TECHNIQUE ----- | 24 |
| C. METHOD OF WEIGHTED RESIDUALS (GALERKIN'S METHOD) ----- | 29 |
| III. ANALYSIS OF CONVECTIVE HEAT TRANSFER BETWEEN PARALLEL PLATES ----- | 37 |
| IV. THEORETICAL RESULTS ----- | 41 |
| A. FINITE-ELEMENT ANALYSIS OF CONTINUUM ----- | 41 |
| 1. Discretization of the Continuum ----- | 41 |
| 2. Selection of Interpolation Functions -- | 43 |
| 3. Determination of the Elemental Properties ----- | 45 |
| B. DERIVATION OF ELEMENT MATRICES ----- | 51 |
| C. STRUCTURE OF COMPUTER PROGRAMS FOR FLOW ANALYSIS ----- | 62 |
| D. NUMERICAL RESULTS ----- | 71 |

| | | |
|---------------------------|--|-----|
| V. | EXPERIMENTAL PROCEDURE ----- | 74 |
| A. | ARRANGEMENT OF TEST APPARATUS ----- | 74 |
| B. | HOLOGRAPHIC INTERFEROMETRY APPLICATIONS --- | 80 |
| VI. | CONCLUSIONS ----- | 84 |
| APPENDIX A | FIGURES ----- | 86 |
| APPENDIX B | BRIEF REVIEW ON CALCULUS OF VARIATIONS - | 98 |
| COMPUTER OUTPUT | ----- | 102 |
| A. | COUETTE FLOW PROBLEM ----- | 102 |
| B. | STEADY STATE HEAT TRANSFER PROBLEM ----- | 108 |
| COMPUTER PROGRAMS | ----- | 117 |
| A. | COUETTE FLOW PROBLEM ----- | 117 |
| B. | STEADY STATE FLUID MECHANICS PROBLEM ----- | 134 |
| C. | TIME-DEPENDENT FLUID MECHANICS PROBLEM ----- | 153 |
| LIST OF REFERENCES | ----- | 193 |
| INITIAL DISTRIBUTION LIST | ----- | 195 |

NOMENCLATURE

| | |
|----------|--|
| C_p | - specific heat of fluid |
| D | - width of the enclosure |
| Δ | - area of element triangle |
| g | - gravitational acceleration |
| Gr_L | - Grashof number in L direction = $\frac{gBL^3(T_H-T_C)}{\nu^2}$ |
| L | - height of the enclosure |
| L_i | - natural coordinates |
| N_i | - interpolation functions |
| P | - pressure (either wall or fluid) |
| Pr | - Prandtl number = $\frac{\nu}{\alpha}$ |
| Ra_D | - Rayleigh number in D direction = $\frac{gBD^3(T_H-T_m)}{\nu}$ |
| Ra_L | - Rayleigh number in L direction = $\frac{gBL^3(T_H-T_C)}{\nu}$ |
| T | - temperature (either wall or fluid) |
| t | - time relative to beginning of solution |
| T_C | - temperature at cold wall |
| T_H | - temperature at hot wall |
| T_m | - mean temperature of fluid = $\frac{(T_H+T_C)}{2}$ |
| u | - velocity in x-direction |
| v | - velocity in y-direction |

- x - independent coordinate in horizontal direction
- y - independent coordinate in vertical direction
- α - thermal diffusivity of fluid = $\frac{\kappa}{\rho C_p}$
- β - coefficient of thermal expansion of fluid
- κ - thermal conductivity of fluid
- μ - dynamic viscosity of fluid
- ν - kinematic viscosity of fluid = $\frac{\mu}{\rho}$
- ρ - fluid density
- $\Omega^{(e)}$ - domain of integration for element (e)

I. INTRODUCTION

A. HOLOGRAPHIC INTERFEROMETRY

The phenomenon of interference has had a considerable influence on the development of physics. Thomas Young's observation and explanation of the interference of the beams through two holes provided the basis for Fresnel's wave theory of light and the same experiment has been used as the foundation of modern coherence theory.

Derived from interference is the technique of interferometry, now one of the important methods of experimental physics. The father of visible-light interferometry was A. A. Michelson, who was awarded in 1907 the Nobel prize in physics for "his optical instruments of precision and the spectroscopic and metrological investigations he has executed with them." Applications to other spectral regions were more recent: the first use of interferometry in radio astronomy was reported in 1947, and infra-red interference spectroscopy was successfully employed some thirteen years later.

Ever since the wave nature of light was generally accepted, interferometry has been the primary method for making measurements with great accuracy. The very small

wavelength of light, on the order of 5×10^{-5} cm, and the fact that interferometric means are available for detecting changes of only a small fraction of this length, indicates the degree of accuracy which can be achieved. The widespread applications of the method attest to its general usefulness. Interferometry is used for testing optical components, optical gauging of machine tools, studying air flow in wind tunnels, and standardizing the fundamental units of length. Therefore it is understandable that any fundamental improvement or innovation in this interferometric technique would find many applications over a wide field.

Holographic interferometry is just such an innovation. Holography may be described as a photographic technique in which the amplitude and phase characteristics emanating from a coherent light source are recorded and later reproduced. This reproduction assumes the form of a three-dimensional image of the original subject. Holography has widened the scope of interferometry to such a degree that holographic interferometry is now considered a standard tool in engineering laboratories all over the world.

Conventional interferometry can be utilized to make measurements on highly polished surfaces of relatively simple shape. Holographic interferometry extends this

range by allowing measurements to be made on three-dimensional surfaces or arbitrary shape and condition. A roughly processed machine part can now be measured to optical tolerance. Furthermore, with the holographic technique a complex object can be examined interferometrically from many different perspectives, because of the three-dimensional nature of the hologram. A single interferometric hologram is equivalent to many observations with a conventional interferometer. This property is especially useful for observations of such things as fluid flow in a wind tunnel. A third departure of holographic interferometry from conventional interferometry is that an object can be interferometrically examined at two different times; one can detect with wavelength accuracy any changes undergone by an object over a period of time. The present object can thus be compared with itself as it was at an earlier time. This is a great advantage in many fields. For example, a large lens can be tested before and after mounting. Similarly, with the aid of pulsed lasers, a machine part can be interferometrically compared with itself statically as well as dynamically.

Methods of holographic interferometry include single- and double-exposure as well as pulsed laser interferometry.

In this thesis, only single-exposure holographic interferometry was considered since it corresponds to real-time interferometry, that is, a method which allows one to observe changes in a subject as they actually occur.

B. CONCEPT AND HISTORY OF THE FINITE-ELEMENT METHOD

One must often resort to numerical procedures in order to obtain quantitative approximate solutions to linear and nonlinear problems in continuum mechanics. However, regardless of the initial assumptions and the methods used to formulate a problem, if numerical methods are employed in evaluating the results, the continuum is, in effect, approximated by a discrete model in the solution process. This observation suggests a logical alternative to the classical approach, namely, represent the continuum by a discrete model at the onset. One such approach, based on the idea of piecewise approximating continuous fields, is referred to as the finite-element method. Its simplicity and generality make it an attractive candidate for applications to a wide range of engineering problems.

Classically, the analysis of continuous systems often began with investigations of the properties of small differential elements of the continuum under investigation. Relationships were established among mean values of various

quantities associated with the infinitesimal elements, and partial differential equations or integral equations governing the behavior of the entire domain were obtained by allowing the dimensions of the elements to approach zero as the number of elements became infinitely large.

In contrast to this classical approach, the finite-element method begins with investigations of the properties of elements of finite dimensions. The equations describing the continuum may be employed in order to arrive at the properties of these elements, but the dimensions of the elements remain finite in the analysis, integrations are replaced by finite summations, and the partial differential equations of the continuous media are replaced, for example, by systems of algebraic or ordinary differential equations. The continuum with infinitely many degrees of freedom is thus represented by a discrete model possessing a finite number of degrees of freedom. Moreover, if certain completeness conditions are satisfied, then, as the number of finite elements is increased and their dimensions are decreased, the behavior of the discrete system converges to that of the continuous system. A significant feature of this procedure is that, in principle, it is applicable to the analysis of finite deformations of materially

nonlinear, nonhomogeneous bodies of any geometrical shape with arbitrary boundary conditions.

The practice of representing a structural system by a collection of discrete elements dates back to the early days of aircraft structural analysis, when wings and fuselages, for example, were treated as assemblages of stringers, skins, and shear panels. By representing a plane elastic solid as a collection of discrete elements composed of bars and beams, Hennikoff [1941] introduced his "framework method," a forerunner to the development of general discrete methods of structural mechanics. Topological properties of certain types of discrete systems were examined by Kron [1939], who developed systematic procedures for analyzing complex electrical networks and structural systems. Courant [1943] presented an approximate solution to the St. Venant torsion problem in which he approximated the warping function linearly in each of an assemblage of triangular elements and proceeded to formulate the problem using the principle of minimum potential energy. Courant's piecewise application of the Ritz method involves all the basic concepts of the procedure now known as the finite-element method. In 1954, Argyris and his collaborators began a series of papers in which they developed certain

generalizations of the linear theory of structures and presented procedures for analyzing complicated discrete structural configurations in forms easily adapted to the digital computer.

The formal presentation of the finite-element method together with the direct stiffness method for assembling elements was attributed to Turner, Clough, Martin, and Topp [1956], who employed the equations of classical elasticity to obtain properties of a triangular element for use in the analysis of plane stress problems. It was Clough [1960], who first used the term "finite elements" in a later paper devoted to plane elasticity problems.

Concepts of the method became more understandable after 1963 when Besseling [1969], Melosh [1970], Fraeys de Veubeke [1971], and Jones [1972] recognized that the finite-element method was a form of the Ritz technique and demonstrated its generality for handling elastic continuum problems. In 1965, the finite-element method received an even broader interpretation when Zienkiewicz and Cheung [1973] reported that it was applicable to all field problems which could be cast into variational form. During the late 1960's and early 1970's, while mathematicians were working on establishing errors, bounds, and convergence

criteria for finite-element approximations, engineers and other appliers of this same method were also studying similar concepts for various problems in the area of solid mechanics.

Although a major portion of the literature written to date on the finite-element method deals with static and dynamic structural analysis, there has been a continuing steady increase in the number of applications in other fields. The goal of this thesis was to develop a computer program, utilizing the finite-element method, which could accurately analyze laminar natural convection within a vertical rectangular enclosure. The program should be able to properly analyze axisymmetric as well as two-dimensional flows.

II. FUNDAMENTAL THEORY OF FINITE-ELEMENT ANALYSIS

In this section the fundamental theory on which the thesis was based is presented. Highlighted topics include the variational principle, some basic concepts of finite-element analysis and the Ritz technique, and finally the method of weighted residuals featuring the Galerkin criterion. The variational principle and the Galerkin method are looked at in detail in regards to the derivation of finite-element equations.

The finite-element method envisions a solution region as built up of many small, interconnected subregions or elements. Such a model of a problem gives a piecewise approximation to the governing equations. The basic premise of the finite-element method is that a solution region can be analytically modeled or approximated by replacing it with an assemblage of discrete elements. These finite-element discretization procedures reduce the problem to one of a finite number of unknowns by dividing the solution region into elements and by expressing the unknown field variable in terms of assumed approximating or interpolation functions within each element. The interpolation functions

are defined in terms of the value of the field variables at specified points called nodes or nodal points. Nodes usually lie on the element boundaries where adjacent elements are considered to be connected. In addition to boundary nodes, an element may also have a few interior nodes (although this was not the case in the choice of linear and quadratic triangular elements utilized in this thesis). The nodal values of the field variable and the interpolation functions for the elements completely define the behavior of the field variable within the elements. For the finite-element representation of a particular problem, the nodal values of the field variable become the new unknowns. Once these unknowns are found, the chosen interpolation functions define the field variable throughout the assemblage of elements.

Clearly, the nature of the solution and the degree of approximation depend not only on the size and number of the elements used, but also on the interpolation functions selected. As one would expect, functions cannot be arbitrarily chosen since certain compatibility conditions must be satisfied. Often such functions are selected so that the field variable and/or its derivatives are continuous across adjoining element boundaries. Another important

feature of the finite-element method which sets it apart from other approximate numerical methods is its ability to formulate solutions for individual elements before putting them together to represent the entire problem.

The finite-element method has gained much popularity and has been utilized extensively in recent years because it has, in general, several outstanding advantages. These are the following:

1. Non-homogeneous configurations may be treated with relative simplicity.
2. The elements can be graded in size and shape to follow boundaries of arbitrary shape.
3. Once a computer program has been developed, problems of the same variety can be solved simply by supplying the computer with appropriate data.

There are at least three distinct approaches one may employ in order to obtain finite-element equations of a particular system. In order of increasing versatility they are: (1) the direct approach, (2) the variational principle, and (3) the weighted residuals approach.

The direct approach can be used only for relatively simple problems in which discrete elements may be easily identified. Once these elements have been selected, direct

physical reasoning is introduced to establish the element equations in terms of pertinent variables. The final step is then to combine the element equations to form the governing equations of the complete system.

A detailed explanation of the remaining two approaches will be given in Subsections A, B and C to follow.

Whichever one of these three particular approaches is utilized, the finite-element method follows a systematic step-by-step process when applied to continuum problems. They are:

1. Discretize the continuum.

The entire flow region under study is divided into a series of subregions or elements assumed to be interconnected at a finite number of nodal points; thus a program originally exhibiting an infinite number of degrees of freedom is made finite. The elements used can be triangular, rectangular, or almost any shape. Also, information must be fed into a computer giving global coordinates of the nodes and topology of the system.

Finally, selection of which field variables are to be used to satisfactorily describe solution domain must be indicated at this point in the process.

2. Select interpolation functions, $N_i^{(e)}$

From the nodal values one represents the value of the field variable over the element by means of interpolation functions. Often, although not always, polynomials are selected as these functions because they are easy to integrate and differentiate. The number of nodes and the order of the interpolation polynomials are interrelated. The field variable itself may be a scalar, a vector, or a higher-order tensor.

3. Find the element properties.

Essentially, the problem is solved at the element level. The matrix equations expressing the properties of the individual elements are determined. This can be accomplished by any one of the three approaches previously mentioned: the direct method, the variational principle, or the weighted residual method. The approach used depends entirely on the nature of the particular problem.

4. Assemble the element properties to obtain the system equations.

In this step, one combines the matrix equations expressing the behavior of the elements to form the matrix equations expressing the behavior of the entire solution region or system. The matrix equations for the system

exhibit the same form as the equations for an individual element except that they contain many more terms because they include all the nodes. The basis for this assembly procedure stems from the fact that, at a node where elements are interconnected, the value of the field variable is the same for each element sharing that node.

5. Solve the system equations.

From the previous step, a set of simultaneous equations are derived which can now be solved to obtain the unknown nodal values of the field variable. If these equations are linear, a number of standard solution techniques may be employed; if the equations are nonlinear, their solution is more difficult to obtain, but several alternative approaches do exist that lead to satisfactory results.

6. Make additional computations if desired.

It may be desired to use the solution of the system equations to calculate other important parameters, i.e., from the nodal values of the pressure, one might wish to calculate velocity distributions.

It is worth making mention of the fact that several of the steps in the above process are essentially the same regardless of the type of problem (this thesis was devoted

to the fluid mechanics problem). Thus, only steps three (3) and six (6) might differ for any given situation, in that the equations describing the elements could vary. The other steps would be the same. This generality of the finite-element method is, without doubt, one of its greatest strengths.

A. VARIATIONAL PRINCIPLE

Often, continuum problems have different but yet equivalent formulations, such as a differential formulation and a variational formulation. In the differential case, the problem is to integrate a differential equation or a system of differential equations subject to given boundary conditions. In the classical variational formulation, the problem is to find the unknown function or functions which extremize (maximize, minimize) or make stationary a functional or system of functionals subject to the same specified boundary conditions. The two problem formulations are equivalent because the functions that satisfy the differential equations and their boundary conditions also extremize or make stationary the functionals. This equivalence is apparent from the calculus of variations, which shows that the functionals are extremized or made stationary only when

one or more Euler equations and their boundary conditions are satisfied. Consequently, these equations are precisely the governing differential equations of the problem. To illustrate this duality concept, Appendix B provides a brief review and introduction to some basic ideas of the calculus of variations.

B. FINITE-ELEMENT METHOD AND THE RITZ TECHNIQUE

The Ritz technique is basically a procedure for transforming a continuous medium into an approximated lumped parameter system. A more qualitative definition would be that the Ritz method consists of assuming the form of the unknown adjustable parameters. From this family of trial or coordinate functions, that particular function which renders the functional stationary is then selected. The procedure is to substitute the trial functions into the functional and thereby express the functional in terms of the adjustable parameters. This functional is then differentiated with respect to each parameter, and the resulting equation is set equal to zero. If there are n unknown parameters, there will be n simultaneous equations to be solved for these parameters. By this means, the approximate solution is chosen from the family of assumed solutions.

This procedure does nothing more than give one the "best" solution from the family of assumed solutions. Clearly, then, the accuracy of the approximate solution depends on the choice of trial functions. These trial functions are required to be defined over the whole solution domain and must satisfy at least some and usually all of the boundary conditions. Sometimes, if the general nature of the desired solution is known, the approximation can be improved by choosing the trial functions to reflect this nature. If, by chance, the exact solution is contained in the family of trial solutions, then the Ritz technique gives the exact solution as expected. Generally, the approximation improves as the size of the family of trial functions and the number of adjustable parameters increase. If the trial functions are part of an infinite set of functions that are capable of representing the unknown function to any degree of accuracy, the process of including more and more trial functions leads to a series of approximate solutions which converge to the true solution. Often a family of trial functions is constructed from polynomials of successively increasing degree.

To illustrate the Ritz technique, consider the following simple example. Suppose it is desired to find the general

function $\phi(x)$ satisfying

$$\frac{d^2\phi}{dx^2} = -f(x)$$

with boundary conditions of $\phi(a)=A$ and $\phi(b)=B$ specified.

It is assumed that $f(x)$ is a continuous function in the closed interval $[a,b]$. This problem is equivalent to finding the function $\phi(x)$ that minimizes the functional

$$I(\phi) = \int_b^a \left[\frac{1}{2} \left(\frac{d\phi}{dx} \right)^2 - f(x)\phi(x) \right] dx$$

which is of the form $I(\phi) = \int_{x_1}^{x_2} F(x, \phi, \phi_x, f(x)) dx$

Ignoring the fact that this problem possesses an exact solution, we will attempt to find an approximate solution. According to the Ritz method, the desired solution can be assumed to be approximately represented in $[a,b]$ by a combination of selected trial functions of the form

$$\phi(x) = C_1 \psi_1(x) + C_2 \psi_2(x) + \dots + C_n \psi_n(x), \quad a \leq x \leq b$$

where the n constants C_i are the adjustable parameters to be determined. The trial functions should be selected so that the expression for $\phi(x)$ satisfies the boundary conditions regardless of the choice of the constants C_i . Using

polynomials is a simple and convenient way of constructing the trial functions. Therefore

$$\phi(x) \approx (x-a)(x-b)(C_1 + C_2 x + C_3 x^2 + \dots + C_n x^{n-1})$$

is a possible series of trial functions. When this approximate expression for $\phi(x)$ is substituted into the functional to be minimized, and after the integration has been carried out, the functional is of the form

$$I = I(C_1, C_2, \dots, C_n).$$

Since the C_i are required to be chosen such that they minimize I , employing differential calculus, the following partial differential equations are formulated

$$\frac{\partial I}{\partial C_1} = 0, \frac{\partial I}{\partial C_2} = 0, \dots, \frac{\partial I}{\partial C_n} = 0$$

These n equations are then solved for the n parameters C_i , and the accuracy of the approximate solution depends on the number of C 's used in the trial function. Generally, as n increases the accuracy improves. To assess the improvement in accuracy as more C 's are utilized, the problem is solved repeatedly by taking successively more terms in the approximation, that is

$$\phi_1(x) \approx (x-a)(x-b)C_1$$

$$\phi_2(x) \approx (x-a)(x-b)(C_1+C_2x)$$

$$\phi_3(x) \approx (x-a)(x-b)(C_1+C_2x+C_3x^2)$$

and so on. By comparing the results at the end of each calculation, the effect on accuracy of adding more terms can be estimated.

The finite-element method and the Ritz technique are essentially equivalent. Each method uses a set of trial functions as the starting point for obtaining an approximate solution; both methods take linear combinations of these trial functions; and both methods seek the combination of trial functions that renders a given functional stationary. The major difference between these approximating methods stems from the fact that the assumed trial functions in the finite-element method are not defined over the entire solution domain, and they must satisfy not just any boundary conditions, but only certain continuity conditions and then only sometimes. Since the Ritz technique uses functions construed over the whole domain, it can be employed only for domains of relatively simple geometric shape. Also, these trial functions associated with the Ritz method are required to satisfy at least some and usually all of the

boundary conditions. In the finite-element method the same geometric limitations exist, but only for the elements. Due to the fact that elements with simple shapes can be assembled to represent exceedingly complex geometries, the finite-element method is a far more versatile tool than the Ritz technique. From a strict mathematical standpoint, the finite-element method is a special case of the Ritz technique only when the piecewise trial functions obey certain continuity and completeness conditions that are stipulated over just the element alone.

C. METHOD OF WEIGHTED RESIDUALS (GALERKIN'S METHOD)

The third and final approach to the finite-element method involves a procedure that is more generalized and straightforward than either of its two predecessors.

The relationship between the well-known Ritz technique and the finite-element method enables one to view the finite-element discretization procedure as simply another means for finding approximate solutions to variational problems. In fact, these finite-element equations were shown to be derived by requiring that a given functional be stationary. This broad variational interpretation is the one most widely used to derive element equations, and it is the

most convenient approach whenever a classical variational statement exists for a given problem.

However, applied scientists and engineers encounter practical problems for which classical variational principles are unknown. In these cases finite-element techniques are still applicable, but more generalized procedures characteristic of the method of weighted residuals must be employed to derive the element equations. Through certain generalizations, finite-element equations may be derived directly from the governing differential equations of the problem without reliance on any classical, quasi-variational, or restricted variational "principles." This procedure allows one to apply the finite-element method to almost all practical problems of mathematical physics.

The method of weighted residuals is a technique for obtaining approximate solutions to linear and nonlinear partial differential equations. It offers still another means with which to formulate the finite-element equations. Applying the method of weighted residuals involves basically two steps. The first step is to assume the general functional behavior of the dependent field variable in some way so as to approximately satisfy the given differential equation along with its associated boundary conditions.

Substitution of this approximation into the original differential equation and boundary conditions then results in some error called a residual. This residual is required to vanish in some average sense over the entire solution domain. The second step entails solving the equation(s) resulting from step one and thereby specializing the general functional form to a particular function, which in turn becomes the approximate solution sought.

To be more specific, the following typical problem is offered. Suppose it is desired to find an approximate functional representation for a general field variable ϕ governed by the differential equation

$$\mathcal{L}(\phi) - f = 0 \quad (2.1)$$

in the domain D bounded by the surface Σ . \mathcal{L} is a linear or nonlinear differential operator and the function f is a known function of the independent variables. Also, proper boundary conditions are assumed to be prescribed on Σ . The method of weighted residuals is now applied in two steps. First, the unknown exact solution ϕ is approximated by $\hat{\phi}$, where either the functional behavior of $\hat{\phi}$ is completely specified in terms of unknown parameters, or the functional dependence on all but one of the independent variables is given while the functional dependence on the

remaining independent variable is left unspecified. Thus the dependent variable is approximated by

$$\phi \approx \hat{\phi} = \sum_{i=1}^m N_i C_i \quad (2.2)$$

where the N_i are the assumed functions and the C_i are either the unknown parameters or unknown functions of one of the independent variables. The m functions N_i are usually chosen to satisfy the global boundary conditions of the system in question. When $\hat{\phi}$ is substituted into equation 2.1, it is unlikely that this equation will not be satisfied, that is,

$$\mathcal{L}(\hat{\phi}) - f \neq 0$$

but in fact,

$$\mathcal{L}(\hat{\phi}) - f = e$$

where e is the residual or error that results from approximating ϕ by $\hat{\phi}$. The method of weighted residuals seeks to determine the m unknowns C_i in such a way that the error e over the entire solution domain is small. This is accomplished by forming a weighted average of the error and specifying that this weighted average vanish over the solution domain. In other words, m linearly independent weighting functions, W_i , are chosen such that

$$\int_D [\mathcal{L}(\hat{\phi}) - f] W_i dD = \int_D e W_i dD = 0, \quad i=1,2,\dots,m \quad (2.3)$$

The form of the error distribution principle expressed in equation 2.3 depends on the choice of weighting functions. Once these are specified, equation 2.3 represents of a set of m equations, which may be either algebraic or ordinary differential. The second step is to solve for the C_i 's and hence obtain an approximate representation of the unknown general field variable ϕ via equation 2.2. There are many linear problems and even some nonlinear problems for which it can be shown that, as $m \rightarrow \infty$, $\hat{\phi} \rightarrow \phi$, but, in general, studies of convergence and error bounds are scarce.

Due to the broad choice of weighting functions or error distribution principles than can be used, a variety of weighted residual techniques are likewise available. The error distribution principle most often utilized to derive finite-element equations in the field of aeronautics is known as the Galerkin criterion, or Galerkin's method. Here, the weighting functions are chosen to be the same as the approximating functions employed to represent ϕ , that is, $W_i = N_i$ for $i=1,2,\dots,m$. Therefore Galerkin's method requires that

$$\int_D [\mathcal{L}(\hat{\phi}) - f] N_i dD = 0 \quad (2.4)$$

In the preceding section pertaining to the Ritz technique,

it was assumed that the entire solution domain was being dealt with. However, because equation 2.1 holds for any point in this region, it also holds for any collection of points defining an arbitrary subdomain or element of the whole domain. Consequently, attention may be focused directly on an individual element by means of a local approximation analogous to equation 2.2, but being defined as valid for only one element at a time. Now the finite-element representations of a general field variable become available. The functions N_i become what are known as the interpolation functions $N_i^{(e)}$ defined over the element, and the C_i are the undetermined parameters, which may be the nodal values of the field variable or its derivatives. Then, from Galerkin's method, the equations governing the behavior of an element of the solution domain may be written as

$$\int_D \left[\mathcal{L}(\phi^{(e)}) - f^{(e)} \right] N_i^{(e)} dD^{(e)} = 0, \quad i=1,2,\dots,r \quad (2.5)$$

where, as before, the superscript (e) restricts the range to one element, and

$$\phi^{(e)} = \left[N^{(e)} \right] \{ \phi \}^{(e)}$$

$f^{(e)}$ = forcing function defined over element (e)

r = number of unknown parameters assigned to the element.

There exists a set of equations similar to equation 2.5 for each element of the whole assemblage. Prior to assembling the system equations from the individual element equations, it is required that the choice of approximating functions N_i guarantee the interelement continuity along the boundary necessary for the assembly process. If the field variable is continuous at element interfaces, then C^0 continuity exists; if, in addition, first derivatives of the variable are continuous, C^1 continuity is said to occur; if second derivatives are also continuous, a region of C^2 continuity exists; and so on. This is the standard definition and notation utilized for expressing the degree of continuity of a field variable at element junctions. The higher the order of continuity required in the solution, the narrower one's choice of interpolation functions becomes.

With the above definition of continuity in mind, the compatibility and completeness requirements for such interpolation functions may be stated. If the functions appearing under the integrals in the element equations contain derivatives up to the $(n+1)$ th order, then the following stipulations must be satisfied for assurance of convergence as the element size decreases.

Compatibility requirement: At element interfaces, C^n continuity must exist.

Completeness requirement: Within an element, C^{n+1} continuity must exist. These requirements hold regardless of whether the element equations (integral expressions) were derived using the variational technique or the Galerkin method. For this thesis, n was taken to have a value of zero.

Integration by parts is a convenient way to introduce the natural boundary conditions that must be satisfied on some portion of the system exterior or boundary. Although the boundary terms containing these imposed conditions appear in the equations for each element, during the assembly of the element equations only the boundary elements give nonvanishing contributions. After the assembly process has been completed, the fixed boundary conditions (i.e., specified velocity, pressure or temperature) are conveniently introduced to help simplify the final matrix form of the finite element equation.

III. ANALYSIS OF CONVECTIVE HEAT TRANSFER BETWEEN PARALLEL PLATES

The transfer of heat energy across a fluid layer is accomplished, in general, through the mechanisms of conduction, convection and radiation. This last phenomenon is usually a function of the fluid enclosed between the surfaces and the nature, temperature and configuration of the enclosing boundaries. Radiation takes place independently of the conduction and convection as long as there is no absorption by the fluid, and therefore under these conditions it can be considered separately. The phenomena of conduction and convection are closely interdependent and are usually analyzed together. Buoyancy forces result from differences in density within the fluid and are caused by heat transfer to or from this fluid. Natural convection may then be thought of as fluid motion of the system due to the activation of these buoyance forces. In a two-dimensional plane, such heat transfer across a vertical, enclosed fluid layer is a function of the Grashof number, the Prandtl number and the fluid layer height-to-width ratio (L/D).

Natural convection plays a very important role in materials processing at high temperatures where agitation by other means is impracticable, or where the existence of temperature gradients is an inherent characteristic of the system.

The steady convective motion of a lubricating fluid contained within a long, rectangular enclosure was investigated. Holographic interferometry and numerical approximation were the experimental and theoretical analysis tools, respectively.

The two vertical walls of the enclosure were held at different temperatures, and the top and bottom were deemed perfect insulators (Figure 1). It was considered that the length of the enclosure (7 inches) was sufficiently long in the direction normal to the plane of Figure 1 for the motion to be assumed two-dimensional. Another assumption made was that the fluid motion was laminar. Experimental evidence indicates that such an assumption is valid provided the Rayleigh number based on cavity height is less than about 10^8 (Ra_L in this study was calculated to be 1.018×10^7). Using this value and a value of the Prandtl number of 1.0755×10^4 , determined from the ratio of kinematic viscosity to thermal diffusivity of the fluid, a system

Grashof number of 946.4 was calculated. The temperatures of the vertical walls $x=0$ and $x=D$ were defined to be T_H and T_C respectively. If $(T_H - T_C)$ in degrees Fahrenheit is sufficiently small with respect to T_C , the Boussinesq approximation may be introduced which neglects density variations in inertia terms of the equations of motion, but retains it in the buoyancy term. One final assumption was made that all other relevant thermodynamic and transport properties were independent of temperature and that compressibility and viscous dissipation effects were negligible.

The problem now was to find the time and spatial dependence of the velocities and the temperatures within the system.

The governing differential equations expressing conservation of mass, momentum (both in x- and y-directions) and energy were

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (3.1)$$

$$\begin{aligned} \frac{\partial u}{\partial t} + u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = \frac{\mu}{\rho} \left(\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} \right) - \frac{1}{\rho} \frac{\partial P}{\partial x} \\ + gB(T - T_m) \end{aligned} \quad (3.2)$$

$$\frac{\partial v}{\partial t} + u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = \frac{\mu}{\rho} \left(\frac{\partial^2 v}{\partial x^2} + \frac{\partial^2 v}{\partial y^2} \right) - \frac{1}{\rho} \frac{\partial P}{\partial y} \quad (3.3)$$

$$\frac{\partial T}{\partial t} + u \frac{\partial T}{\partial x} + v \frac{\partial T}{\partial y} = \alpha \left(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2} \right) \quad (3.4)$$

The solution to the foregoing set of dynamic equations must satisfy the following boundary conditions on the walls,

$u_0 = v_0 = 0$, no velocity on any of the four walls

$T = T_H$ or T_C given on the two vertical walls

$P = P_{ATMOS}$, also given on the two vertical walls.

IV. THEORETICAL RESULTS

A. FINITE-ELEMENT ANALYSIS OF THE CONTINUUM

1. Discretization of the Continuum

Since the fundamental premise of the finite-element method is that a continuum or solution domain of arbitrary shape can be accurately modeled by an assemblage of simple shapes, most finite elements are geometrically simple also. This statement especially pertains to the choice of the triangular-shaped element which would represent the unknown system parameters in this study, that is, the velocity, temperature and pressure. The main reason behind this choice was the fact that the three-node flat triangular element is the simplest two-dimensional element available, and hence an assemblage of triangles could always depict a two-dimensional domain with any number of straight sides. The solution domain in this problem was the vertical rectangular enclosure, a relatively simple-shaped continuum which posed no problem for the triangular elements. Twelve (12) elements were utilized to represent the 8.5 inch by 1.875 inch area. They were interconnected to each other and the boundary at a total of thirty-five (35) nodal points, of which twelve (12) were corner nodes (Figure 2).

Arriving at this figure of thirty-five nodes was not an arbitrary process. Once pressure was chosen to be linearly approximated, system velocities and temperature were required to assume polynomial approximation of one degree higher, or quadratic, if the highest solution accuracy was to be achieved. For triangular elements, a complete n th-order polynomial requires $\frac{1}{2}(n+1)(n+2)$ nodes for its specification. Therefore, a 1st order, or linearly approximated, polynomial is associated with a three-node triangle; and a quadratic polynomial relates to a six-node triangle.

The three-node elements, with their nodes on the corners, may be thought of as being superimposed onto the six-node elements. Such elements contain, in addition to the corner nodes, nodes located at the midpoint of each side of the triangle. Twelve triangular elements of the six-node variety may be interconnected to form the solution domain shown in Figure 2; this domain possessing exactly thirty-five nodes.

Each element (6-node and 3-node) specifies uniquely a complete polynomial of the order necessary to give C^0 continuity, and hence satisfy the completeness and compatibility requirements for elemental assemblage.

Next, the distinction between local and global node-numbering had to be made. Since each element in the triangular mesh had six nodes, the local nodes were identified as such by starting in the upper left hand corner of each element and numbering counterclockwise around the element. The global node system is a method for uniting these independent elements along with their nodes into one distinct entity. Figure 3 summarizes the relation between local and global numbering for four (4) such elements. This figure defines the system topology or the connectivity of the system.

2. Selection of the Interpolation Functions

In the preceding subsection it was mentioned that linear approximation was used for values representing nodal pressures, while both velocity and temperature varied in a quadratic fashion within the elements. Such a relationship was based on the governing equations of the system, in which the highest order of partial differential equations involving pressure was one, while partial derivatives of u , v and T existed up to second order. Therefore, choosing linear pressures required the remaining three nodal parameters or field variables to take on quadratic approximation.

The functions employed to represent the behavior of these field variables within an element are known as interpolation or approximating functions. Their order within an element depends on the number of degrees of freedom assigned to that element. In this study, two different polynomial series were selected as the first and second order interpolation functions. Associated with these series were coefficients made up of generalized coordinates, that is, independent parameters which specified the magnitude of the prescribed distribution for each field variable (u, v, P, T). These polynomials were represented as follows

$$P(x,y)^{(e)} = C_1^{(e)} + C_2^{(e)}x + C_3^{(e)}y \quad (4.1)$$

for the linear pressure terms, and

$$\begin{aligned} \phi(x,y)^{(e)} = & C_1^{(e)} + C_2^{(e)}x + C_3^{(e)}y + C_4^{(e)}x^2 + \\ & C_5^{(e)}xy + C_6^{(e)}y^2 \end{aligned} \quad (4.2)$$

with ϕ being a generalized quadratic field variable (either u, v, or T in this case, and the superscript 'e' standing for element.

The next step in the process was to solve for the generalized coordinate $C_i^{(e)}$ in terms of the as yet unknown field variables. This gave the desired interpolation, but the form of the resulting equations was not convenient. As a final step then, the equations were rearranged until they appeared as

$$P(x,y)^{(e)} = N_1^P(x,y)P_1 + N_2^P(x,y)P_2 + N_3^P(x,y)P_3 = \left[N^P \right] \left\{ P \right\} \quad (4.3)$$

and

$$\phi(x,y)^{(e)} = N_1^\phi(x,y)\phi_1 + N_2^\phi(x,y)\phi_2 + N_3^\phi(x,y)\phi_3 + N_4^\phi(x,y)\phi_4 + N_5^\phi(x,y)\phi_5 + N_6^\phi(x,y)\phi_6 = \left[N^\phi \right] \left\{ \phi \right\} \quad (4.4)$$

where N_i^u , N_i^v , and N_i^T were the specific interpolation functions in equation (4.4) for this study and were all equal in form, i.e., $N_i^u = N_i^v = N_i^T = N_i$.

3. Determination of the Elemental Properties

In this thesis, the Galerkin method was utilized to determine the element properties. This procedure applied at a general node i of an isolated element becomes, in view of equations 3.1-3.4,

$$\int_{\Omega^{(e)}} H_i \left(\frac{\partial u^{(e)}}{\partial x} + \frac{\partial v^{(e)}}{\partial y} \right) dx dy = 0 \quad (4.5)$$

$$\int_{\Omega^{(e)}} W_i \left[\frac{4}{8} \left(\frac{\partial^2 u^{(e)}}{\partial x^2} + \frac{\partial^2 u^{(e)}}{\partial y^2} \right) - \frac{1}{8} \frac{\partial P^{(e)}}{\partial x} + gB(T^{(e)} - T_m) \right. \\ \left. - u^{(e)} \frac{\partial u^{(e)}}{\partial x} - v^{(e)} \frac{\partial u^{(e)}}{\partial y} - \frac{\partial u^{(e)}}{\partial t} \right] dx dy = 0 \quad (4.6)$$

$$\int_{\Omega^{(e)}} W_i \left[\frac{4}{8} \left(\frac{\partial^2 v^{(e)}}{\partial x^2} + \frac{\partial^2 v^{(e)}}{\partial y^2} \right) - \frac{1}{8} \frac{\partial P^{(e)}}{\partial y} - u^{(e)} \frac{\partial v^{(e)}}{\partial x} \right. \\ \left. - v^{(e)} \frac{\partial v^{(e)}}{\partial y} - \frac{\partial v^{(e)}}{\partial t} \right] dx dy = 0 \quad (4.7)$$

$$\int_{\Omega^{(e)}} W_i \left[\alpha \left(\frac{\partial^2 T^{(e)}}{\partial x^2} + \frac{\partial^2 T^{(e)}}{\partial y^2} \right) - u^{(e)} \frac{\partial T^{(e)}}{\partial x} \right. \\ \left. - v^{(e)} \frac{\partial T^{(e)}}{\partial y} - \frac{\partial T^{(e)}}{\partial t} \right] dx dy = 0 \quad (4.8)$$

where $W_i(x,y)$ and $H_i(x,y)$ are the weighting or interpolation functions, which were taken as

$$W_i = N_i \text{ and } H_i = N_i^P = L_i \text{ (natural coordinates).}$$

The inertia terms in equations 4.6, 4.7 and 4.8 considerably increased the degree of difficulty of this fluid flow problem when compared to an incompressible viscous flow without inertia. This is because the above mentioned equations are nonlinear, thereby forcing an iterate procedure to be introduced and repeated until the $u_{n+1}^{(e)}$, $v_{n+1}^{(e)}$, and $T_{n+1}^{(e)}$

values converged to the previous $u_n^{(e)}$, $v_n^{(e)}$, and $T_n^{(e)}$ solutions. The subscript n runs from zero to some positive number at which the field variable passes a convergence test.

Integrating each term of equations 4.5-4.8 by parts, and making use of the approximations of equations 4.3 and 4.4, the following results on an elemental level were obtained

$$\begin{aligned} \int_{\Omega^{(e)}} (N_i^P \frac{\partial [N]}{\partial x} dx dy) \{ u \} \\ + \int_{\Omega^{(e)}} (N_i^P \frac{\partial [N]}{\partial y} dx dy) \{ v \} = 0 \end{aligned} \quad (4.9)$$

$$\begin{aligned} \int_{\Omega^{(e)}} \frac{1}{\rho} \left(\frac{\partial N_i}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial [N]}{\partial y} \right) dx dy \{ u \} \\ - \frac{1}{\rho} \int_{\Omega^{(e)}} \frac{\partial N_i}{\partial x} [N^P] dx dy \{ P \} - gB [N_i] \{ T \} \\ - \int_{\Omega^{(e)}} (N_i [N] dx dy) \left\{ \frac{\partial u}{\partial t} \right\} = \left\{ gBT_m \right\} + \int_C N_i^* X ds \end{aligned} \quad (4.10)$$

$$\begin{aligned} \int_{\Omega^{(e)}} \frac{1}{\rho} \left(\frac{\partial N_i}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial [N]}{\partial y} \right) dx dy \{ v \} \\ - \frac{1}{\rho} \int_{\Omega^{(e)}} \frac{\partial N_i}{\partial y} [N^P] dx dy \{ P \} - \int_{\Omega^{(e)}} (N_i [N] dx dy) \left\{ \frac{\partial v}{\partial t} \right\} \\ = \int_C N_i^* Y ds \end{aligned} \quad (4.11)$$

$$\int_{\Omega^{(e)}} \alpha \left(\frac{\partial N_i}{\partial x} \frac{\partial [N]}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial [N]}{\partial y} \right) dx dy \{ T \} - \int_{\Omega^{(e)}} (N_i [N]) dx dy \left\{ \frac{\partial T}{\partial t} \right\} = \int_C N_i Z^* ds \quad (4.12)$$

where $N_i X^* ds$, $N_i Y^* ds$ and $N_i Z^* ds$ are simply lumped-sum contour integrals that introduce the natural boundary conditions for u , v and T respectively. These integral values were labeled QX, QY, and QZ in the computer program. The last term on the left hand side of equations 4.10-4.12 represents the transient nature of the system.

Finally, the element matrix equations were written by inspection from equations 4.9-4.12 and were of the general form

$$[K]^{(e)} \{ \emptyset \}^{(e)} - [K_t]^{(e)} \{ \dot{\emptyset} \}^{(e)} = \{ R \}^{(e)} \quad (4.13)$$

where the square matrices $[K]^{(e)}$ AND $[K_t]^{(e)}$ are known as stiffness matrices, the column vectors $\{ \emptyset \}^{(e)}$ and $\{ \dot{\emptyset} \}^{(e)}$ are the nodal field variable and time derivative vectors, respectively. The column vector $\{ R \}^{(e)}$ signifies the resultant nodal force vector for the element. In the actual computer program, the following identities were used

$$[K] = [TM], \quad [K_t] = [CD], \quad \{ \emptyset \} = \{ X \}, \quad \{ \dot{\emptyset} \} = \{ \dot{X} \},$$

and $\{ R \} = \{ RHS \}$

and the element matrix equations were

$$\begin{array}{c}
 (3r+s) \times (3r+s) \quad (3r+s) \times 1 \\
 \left[\begin{array}{c|c|c|c}
 r[K_1] & [0] & -\frac{1}{\rho}[K_2]^T & g\beta[I] \\
 \hline
 [0] & r[K_1] & -\frac{1}{\rho}[K_3]^T & [0] \\
 \hline
 -[K_2] & -[K_3] & [0] & [0] \\
 \hline
 [0] & [0] & [0] & \alpha[K_1]
 \end{array} \right] \begin{Bmatrix} \{u\}^e \\ \{v\}^e \\ \{p\}^e \\ \{T\}^e \end{Bmatrix} - \\
 \\
 \left[\begin{array}{c|c|c|c}
 [c_D] & [0] & [0] & [0] \\
 \hline
 [0] & [c_D] & [0] & [0] \\
 \hline
 [0] & [0] & [0] & [0] \\
 \hline
 [0] & [0] & [0] & [c_D]
 \end{array} \right] \begin{Bmatrix} \{\dot{u}\}^e \\ \{\dot{v}\}^e \\ \{\dot{p}\}^e \\ \{\dot{T}\}^e \end{Bmatrix} = \begin{Bmatrix} \{AX\}^e \\ \{QY\}^e \\ \{QZC\}^e \\ \{QZ\}^e \end{Bmatrix} \\
 (3r+s) \times (3r+s) \quad (3r+s) \times 1 \quad (3r+s) \times 1
 \end{array}$$

where, $\{AX\}^e = \{g\beta T_m + QX\}$

In the above assemblage, the individual matrix notation utilized was

$$[K_1] = K_1(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dx dy$$

$$[K_2]^T = K_2^T(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial x} N_j^P \right) dx dy$$

$$[K_3]^T = K_3^T(i,j) = \int_{\Omega^e} \left(\frac{\partial N_i}{\partial y} N_j^P \right) dx dy$$

$$[K_2] = K_2(i,j) = \int_{\Omega^e} \left(\frac{\partial N_j}{\partial x} N_i^P \right) dx dy$$

$$[K_3] = K_3(i,j) = \int_{\Omega^e} \left(\frac{\partial N_j}{\partial y} N_i^P \right) dx dy$$

$$[CD] = CD(i,j) = \int_{\Omega^e} N_i N_j dx dy$$

Also, gBT_m is the u velocity forcing function, and r and s are the number of nodes where velocity (or temperature) and pressure are interpolated at, respectively. In this study, $r=6$ and $s=3$, therefore the element matrices were 21×21 and the element column vectors 21×1 .

Once the matrix equations were compiled or assembled on the element level, assembling these properties to obtain the system equations in matrix form also was a relatively simple operation for the digital computer. In essence, the large square matrices were derived by systematically adding

together the contributions of each individual element matrix, and inserting prescribed nodal variables or boundary conditions where applicable. As was brought out in the theoretical section of this thesis, the final assembly now became a system of ordinary differential equations resembling the same format as equation 4.13, i.e.

$$\left[K \right] \left\{ \phi \right\} - \left[K_t \right] \left\{ \dot{\phi} \right\} = \left\{ R \right\} \quad (4.14)$$

The problem solution was completed when these equations were solved for the nodal parameters $\left\{ \phi \right\}$, subject to the discretized initial conditions.

B. DERIVATION OF ELEMENT MATRICES

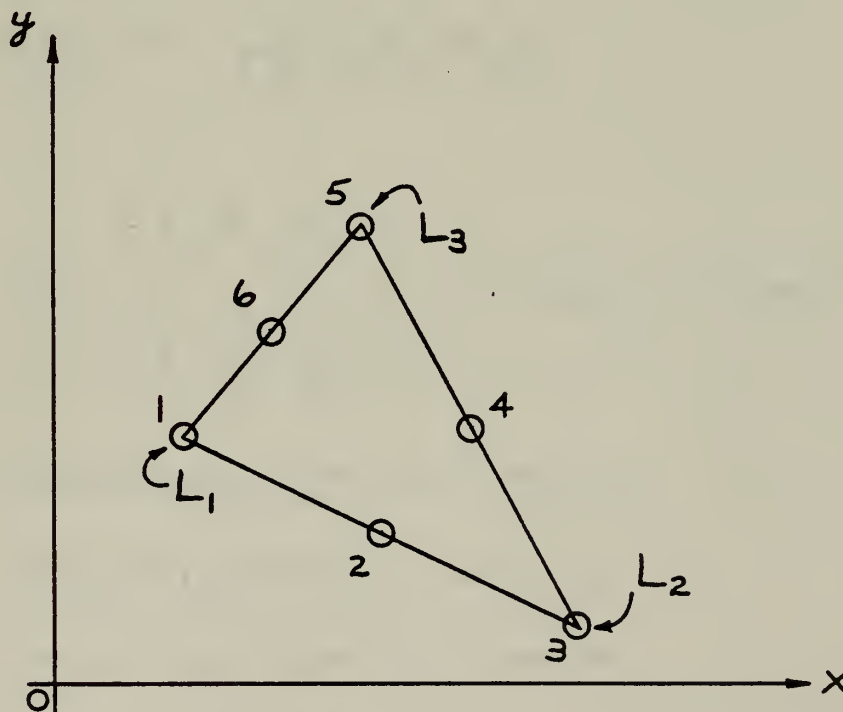
Derivation of various element matrices, referred to previously as simply area integrals over the solution domain, will be discussed in this section. The evaluation of each matrix will be in terms of natural coordinates, that is, weighting functions relating the coordinates of the end nodes to the coordinate of any interior point belonging to the element. The weighting functions are not independent of one another, since their sum must equal unity, i.e.

$$\sum_{i=1}^n L_i = 1 \quad (4.15)$$

where n is the number of external nodes of the element. This expression can be interpreted to mean that one and only one coordinate is associated with node i , having a unit value there and a zero value at every other node. As was previously mentioned in other sections, a general triangular shaped element, such as sketched below, was employed. Then by equation 4.15

$$L_1 + L_2 + L_3 = 1$$

A cartesian coordinate system is used since the fluid flow is assumed to be two-dimensional. Similar results could be derived using cylindrical coordinates for an axisymmetrically shaped element. The original Cartesian coordinates of a



point in the element can now be linearly related to the new natural coordinates by the equations

$$x = L_1 x_1 + L_2 x_2 + L_3 x_3 \quad (4.16)$$

and

$$y = L_1 y_1 + L_2 y_2 + L_3 y_3 \quad (4.17)$$

Solving for the natural coordinates in terms of the Cartesian coordinates gives

$$L_1(x,y) = \frac{1}{2\Delta}(a_1 + b_1 x + c_1 y) \quad (4.18a)$$

$$L_2(x,y) = \frac{1}{2\Delta}(a_2 + b_2 x + c_2 y) \quad (4.18b)$$

and finally

$$L_3(x,y) = \frac{1}{2\Delta}(a_3 + b_3 x + c_3 y) \quad (4.18c)$$

where

$$2\Delta = \begin{vmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{vmatrix} = 2 \text{ (area of triangle 1-2-3)}$$

$$a_1 = x_2 y_3 - x_3 y_2, \quad b_1 = y_2 - y_3, \quad c_1 = x_3 - x_2$$

$$a_2 = x_3 y_1 - x_1 y_3, \quad b_2 = y_3 - y_1, \quad c_2 = x_1 - x_3$$

$$a_3 = x_1 y_2 - x_2 y_1, \quad b_3 = y_1 - y_2, \quad c_3 = x_2 - x_1$$

The interpolation functions N_i for the linear pressures in terms of natural coordinates are merely

$$N_1^P = L_1, N_2^P = L_2, N_3^P = L_3$$

but those interpolation functions that relate to the velocities and temperatures stemming from quadratic approximation possess the form

$$N_1 = 2L_1^2 - L_1$$

$$N_2 = 4L_1L_2$$

$$N_3 = 2L_2^2 - L_2$$

(4.19)

$$N_4 = 4L_2L_3$$

$$N_5 = 2L_3^2 - L_3$$

$$N_6 = 4L_1L_3$$

Another way of envisioning $L_i(x,y)$ for the triangular element is to consider it a ratio of areas. Figure 4 shows how the natural coordinates, often called area coordinates, are related to areas. In this figure, when the point (x_p, y_p) is located on the boundary of the element, one of the area segments vanishes and hence the appropriate area coordinate along that particular boundary is identically zero. For example, if (x_p, y_p) is on line 1-2, then

$$L_3 = \frac{A_3}{\Delta} = 0 \quad \text{since } A_3 = 0$$

There is also a convenient analytical method for integrating area coordinates over the area of a triangular element and involves the formula

$$\int_{A(e)} L_1^\alpha L_2^\beta L_3^\gamma dA(e) = \frac{\alpha! \beta! \gamma!}{(\alpha + \beta + \gamma + 2)!} 2\Delta$$

A summation of values derived using this formula is presented in Table 4.1 for $(\alpha + \beta + \gamma) \leq 4$.

$$\frac{1}{\Delta} \int_{A(e)} L_1^\alpha L_2^\beta L_3^\gamma dA(e) = \frac{A}{B}$$

| $\alpha + \beta + \gamma$ | α | β | γ | A | B |
|---------------------------|----------|---------|----------|----|-----|
| 0 | 0 | 0 | 0 | 1 | 1 |
| 1 | 1 | 0 | 0 | 1 | 3 |
| 2 | 2 | 0 | 0 | 2 | 12 |
| 2 | 1 | 1 | 0 | 1 | 12 |
| 3 | 3 | 0 | 0 | 6 | 60 |
| 3 | 2 | 1 | 0 | 2 | 60 |
| 3 | 1 | 1 | 1 | 1 | 60 |
| 4 | 4 | 0 | 0 | 12 | 180 |
| 4 | 3 | 1 | 0 | 3 | 180 |
| 4 | 2 | 2 | 0 | 2 | 180 |
| 4 | 2 | 1 | 1 | 1 | 180 |

Table 4.1

With this preliminary work finished, the actual derivation of the element matrices may now begin. In all, five matrices will be completely evaluated while one matrix, $[K_1]$, will have only two of its terms derived, due to the extensive amount of time and paper needed to evaluate $[K_1]$ in total. Beginning with this above-mentioned matrix as it appeared in Subsection A,

$$K_1(i,j) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial N_i}{\partial x} \frac{\partial N_j}{\partial x} + \frac{\partial N_i}{\partial y} \frac{\partial N_j}{\partial y} \right) dx dy \quad (4.20)$$

where $\Omega^{(e)}$ is the elemental area representing the solution domain, and i and j both vary from one to six. Since $[K_1]$ is an array multiplying the nodal variables of velocity and temperature, it must be correlated with the quadratic interpolation functions of equation 4.19. For the point (1,1), equation 4.20 becomes

$$K_1(1,1) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial N_1}{\partial x} \frac{\partial N_1}{\partial x} + \frac{\partial N_1}{\partial y} \frac{\partial N_1}{\partial y} \right) dx dy \quad (4.21)$$

where

$$\frac{\partial N_1}{\partial x} = \frac{\partial L_1}{\partial x} (4L_1 - 1) = \frac{b_1}{2\Delta} (4L_1 - 1)$$

and

$$\frac{\partial N_1}{\partial y} = \frac{\partial L_1}{\partial y} (4L_1 - 1) = \frac{c_1}{2\Delta} (4L_1 - 1)$$

substituting these values into equation 4.21

$$\begin{aligned}
K_1(1,1) &= \nu \int_{\Omega^{(e)}} \left[\frac{b_1^2}{4\Delta^2} (4L_1 - 1)^2 + \frac{c_1^2}{4\Delta^2} (4L_1 - 1)^2 \right] dx dy \\
&= \frac{\nu(b_1^2 + c_1^2)}{4\Delta^2} \int_{\Omega^{(e)}} (16L_1^2 - 8L_1 + 1) dx dy
\end{aligned}$$

employing Table 4.1 for these three cases above in which $(\alpha + \beta + \gamma) = 2, 1$, and 0 respectively; plus the relationship that $\int_{\Omega^{(e)}} dx dy = \Delta$

$$K_1(1,1) = \frac{\nu(b_1^2 + c_1^2)}{4\Delta^2} (16 \cdot \frac{2}{12} - 8 \cdot \frac{1}{3} + 1) \Delta$$

or finally

$$K_1(1,1) = \frac{\nu(b_1^2 + c_1^2)}{4\Delta}$$

Next, consider the point $(2,4)$ where

$$K_1(2,4) = \nu \int_{\Omega^{(e)}} \left(\frac{\partial N_2}{\partial x} \frac{\partial N_4}{\partial x} + \frac{\partial N_2}{\partial y} \frac{\partial N_4}{\partial y} \right) dx dy \quad (4.22)$$

where

$$\begin{aligned}
\frac{\partial N_2}{\partial x} &= 4 \left[L_1 \left(\frac{b_2}{2\Delta} \right) + L_2 \left(\frac{b_1}{2\Delta} \right) \right], \\
\frac{\partial N_2}{\partial y} &= 4 \left[L_1 \left(\frac{c_2}{2\Delta} \right) + L_2 \left(\frac{c_1}{2\Delta} \right) \right], \\
\frac{\partial N_4}{\partial x} &= 4 \left[L_2 \left(\frac{b_3}{2\Delta} \right) + L_3 \left(\frac{b_2}{2\Delta} \right) \right], \\
\frac{\partial N_4}{\partial y} &= 4 \left[L_2 \left(\frac{c_3}{2\Delta} \right) + L_3 \left(\frac{c_2}{2\Delta} \right) \right]
\end{aligned}$$

substituting these four values into equation 4.22

$$K_1(2,4) = \nu \int_{\Omega(e)} \left\{ 16 \left[L_1 L_2 \left(\frac{b_2}{2\Delta} \right) \left(\frac{b_3}{2\Delta} \right) + L_2^2 \left(\frac{b_1}{2\Delta} \right) \left(\frac{b_3}{2\Delta} \right) \right. \right. \\ \left. \left. + L_1 L_3 \left(\frac{b_2}{2\Delta} \right)^2 + L_2 L_3 \left(\frac{b_1}{2\Delta} \right) \left(\frac{b_2}{2\Delta} \right) + L_1 L_2 \left(\frac{c_2}{2\Delta} \right) \left(\frac{c_3}{2\Delta} \right) \right. \right. \\ \left. \left. + L_2^2 \left(\frac{c_1}{2\Delta} \right) \left(\frac{c_3}{2\Delta} \right) + L_1 L_3 \left(\frac{c_2}{2\Delta} \right)^2 + L_2 L_3 \left(\frac{c_1}{2\Delta} \right) \left(\frac{c_2}{2\Delta} \right) \right] \right\} dx dy$$

simplifying again, through the use of Table 4.1

$$K_1(2,4) = \frac{\nu}{3\Delta} (b_2 b_3 + c_2 c_3 + 2b_1 b_3 + 2c_1 c_3 + b_1 b_2 + c_1 c_2 + b_2^2 + c_2^2)$$

As can be seen, terms in the K_1 matrix can be quite lengthy and require considerable time to derive. On a more positive note though, this square matrix is symmetric and thus only half the terms need be calculated manually.

Next, attention will be focused on the K_2 and K_3 matrices. These two arrays may be considered together since the only difference between the two of them is that b values are associated with $[K_2]$ and c values with $[K_3]$. Otherwise, they are identical. These two matrices were given as

$$K_2(i,j) = \int_{\Omega(e)} \left(\frac{\partial N_i}{\partial x} N_j^P \right) dx dy \quad (4.23)$$

and

$$K_3(i,j) = \int_{\Omega(e)} \left(\frac{\partial N_i}{\partial y} N_j^P \right) dx dy \quad (4.24)$$

Taking the (3,6), equation 4.23 becomes

$$K_2(3,6) = \int_{\Omega^{(e)}} \left(\frac{\partial N_6}{\partial x} N_3^P \right) dx dy$$

where

$$\frac{\partial N_6}{\partial x} = 4 \left[L_1 \frac{\partial L_3}{\partial x} + L_3 \frac{\partial L_1}{\partial x} \right] \text{ and } N_3^P = L_3, \text{ then}$$

substituting above

$$\begin{aligned} K_2(3,6) &= 4 \int_{\Omega^{(e)}} \left[L_1 \left(\frac{b_3}{2\Delta} \right) + L_3 \left(\frac{b_1}{2\Delta} \right) \right] L_3 dx dy \\ &= \frac{2}{\Delta} \int_{\Omega^{(e)}} (L_1 L_3 b_3 + L_3^2 b_1) dx dy \end{aligned}$$

once again, using Table 4.1

$$K_2(3,6) = \frac{1}{6} (2b_1 + b_3)$$

and consequently

$$K_3(3,6) = \frac{1}{6} (2c_1 + c_3)$$

Following the same procedure throughout each of these 3x6 matrices, complete $[K_2]$ and $[K_3]$ are

$$[K_2] = \frac{1}{6} \begin{bmatrix} b_1 & b_1+2b_2 & 0 & b_2+b_3 & 0 & b_1+2b_3 \\ 0 & 2b_1+b_2 & b_2 & b_2+2b_3 & 0 & b_3+b_1 \\ 0 & b_1+b_2 & 0 & 2b_2+b_3 & b_3 & 2b_1+b_3 \end{bmatrix}$$

and also

$$\begin{bmatrix} K_3 \end{bmatrix} = \frac{1}{6} \begin{bmatrix} c_1 & c_1+2c_2 & 0 & c_2+c_3 & 0 & c_1+2c_3 \\ 0 & 2c_1+c_2 & c_2 & c_2+2c_3 & 0 & c_3+c_1 \\ 0 & c_1+c_2 & 0 & 2c_2+c_3 & c_3 & 2c_1+c_3 \end{bmatrix}$$

The next two matrices to be derived, that is $\begin{bmatrix} K_2 \end{bmatrix}^T$ and $\begin{bmatrix} K_3 \end{bmatrix}^T$, can simply be written down by inspection of the two above arrays. Thus

$$\begin{bmatrix} K_2 \end{bmatrix}^T = \frac{1}{6} \begin{bmatrix} b_1 & 0 & 0 \\ b_1+2b_2 & 2b_1+b_2 & b_1+b_2 \\ 0 & b_2 & 0 \\ b_2+b_3 & b_2+2b_3 & 2b_2+b_3 \\ 0 & 0 & b_3 \\ b_1+2b_3 & b_3+b_1 & 2b_1+b_3 \end{bmatrix}$$

while its counterpart is then

$$[K_3]^T = \frac{1}{6} \begin{bmatrix} c_1 & 0 & 0 \\ c_1+2c_2 & 2c_1+c_2 & c_1+c_2 \\ 0 & c_2 & 0 \\ c_2+c_3 & c_2+2c_3 & 2c_2+c_3 \\ 0 & 0 & c_3 \\ c_1+2c_3 & c_3+c_1 & 2c_1+c_3 \end{bmatrix}$$

Finally, the last elemental matrix to be analyzed is the one associated with the time-dependent nodal parameters. In subsection A this matrix was given as $[CD]$. For convenience here, let $[CD] = [K_t]$, then

$$K_t(i,j) = \int_{\Omega^{(e)}} N_i N_j dx dy \quad (4.25)$$

with both i and j running from one to six. Consider, for example, point (1,5)

$$K_t(1,5) = \int_{\Omega^{(e)}} N_1 N_5 dx dy$$

substituting from equation 4.19

$$\begin{aligned} K_t(1,5) &= \int_{\Omega^{(e)}} (2L_1^2 - L_1)(2L_3^2 - L_3) dx dy \\ &= \int_{\Omega^{(e)}} (4L_1^2 L_3^2 - 2L_1 L_3^2 - 2L_1^2 L_3 + L_1 L_3) dx dy \end{aligned}$$

then from Table 4.1, using $(\alpha + \beta + \gamma)$ four separate times, this term reduces rather easily to

$$K_t(1,5) = - \frac{\Delta}{180}$$

Factoring out a constant of $\frac{\Delta}{180}$, the total K_t matrix takes on the form

$$[K_t] = [CD] = \frac{\Delta}{180} \begin{bmatrix} 6 & 0 & -1 & -4 & -1 & 0 \\ 0 & 32 & 0 & 16 & -4 & 16 \\ -1 & 0 & 6 & 0 & -1 & -4 \\ -4 & 16 & 0 & 32 & 0 & 16 \\ -1 & -4 & -1 & 0 & 6 & 0 \\ 0 & 16 & -4 & 16 & 0 & 32 \end{bmatrix}$$

Which is also a symmetric matrix, thereby allowing faster derivation of the individual terms with less chance of numerical error.

C. STRUCTURE OF COMPUTER PROGRAMS FOR FLOW ANALYSIS

A total of three computer programs analyzing two distinct test cases of fluid flow problems were employed in this thesis. The first was a steady state analysis of Couette flow. This involved determining the solution of the velocity profiles (linear and nonlinear) in a shear- and pressure-induced flow between flat parallel plates.

The upper plate slides in the positive x-direction with a constant velocity u , while the lower plate remains stationary. There is no velocity component normal to the plates, that is, $v=0$ in the y-direction. The governing equations for this particular fluid flow are

Continuity:

$$\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} = 0 \quad (4.26)$$

Momentum:

$$u \frac{\partial u}{\partial x} + v \frac{\partial u}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial x} + \nu \nabla^2 u \quad (4.27)$$

$$u \frac{\partial v}{\partial x} + v \frac{\partial v}{\partial y} = -\frac{1}{\rho} \frac{\partial P}{\partial y} + \nu \nabla^2 v \quad (4.28)$$

For determination of the velocity profile involving only linear terms, the left hand side of equations 4.27 and 4.28 are set equal to zero (no inertia terms).

A physical representation of the Couette flow analyzed is shown in Figure 5. Node and element numbering is the same as in Figure 2. A pressure gradient of -3 units is directed along the x-axis, i.e., $\frac{dP}{dx} = -3$.

The two remaining programs formed the majority of the theoretical portion of this study. They were devised to carry out the calculations for the analysis of two-dimensional or axisymmetric natural convection heat transfer problems.

The first is the lesser-complex steady state approach, whereby all transient conditions characteristic of the system are assumed to have died out, leaving only those steady state parameters remaining to be solved for. Once the finite element matrix equations describing the system are correctly assembled, a library subroutine (LEQT2F) functioning as a linear equation solver is called and the desired nodal parameters can be calculated. The second program takes into account the previously-neglected time dependence of the system by introducing a type of finite difference integration scheme to solve the transient portion of the governing equations. This integration technique must be an iterative procedure in order to circumvent the problem of nonlinearity similar to that resulting from the addition of inertia terms. Furthermore, the integral is solved at successive time steps, with time being increased until the value of the field variable converges, within tolerance, to that of its steady state counterpart. This segment of the total equation is then algebraically combined with the remaining steady state solution to yield values of nodal field variables with improved accuracy. Gravity acting in the longitudinal direction was taken into account for both the two-dimensional and axisymmetric flows.

The flow region to be studied is first defined, followed by the setting up of coordinate axes (Figure 1). The location of the origin of these axes is in most cases arbitrary, except that for a problem involving axial symmetry the x-axis must coincide with the system axis of symmetry. The flow region is then divided into a mesh of triangular elements, and the nodal points are numbered in the sequence previously described. Once the setting-up of the solution domain is completed, computer analysis of the system with its included boundary conditions can be initiated.

Structure of the steady state fluid mechanics problem will be discussed in detail, primarily because it comprises one entire program and with the addition of the transient stiffness matrix elements, accounts for the main program in the time-dependent study.

The program was coded in FORTRAN IV language and begins with a series of DIMENSION statements, which set up the arrays needed in the calculations. As indicated in statements 0260-0310, storage has been allocated for problems with up to 117 nodes; however larger problems can be considered by simply increasing the dimensions of these matrices. The limit of problem size is dictated by the

core storage of the available machine. Next, the program calls for a declaration of the type of program to be solved (either two-dimensional or axisymmetric); then the appropriate problem label is printed (statements 0360-0410). Before proceeding to input the data describing the finite element mesh, all the matrix arrays must be initialized by setting all terms in these arrays equal to zero.

Statements 0930-1150 read into the program the node numbers and the coordinates of the nodes for the complete finite element mesh. Also, the system topology, the element numbers, and the numbers of the six nodes associated with each element are read. Beginning with statement 1300, the velocity, pressure and temperature conditions within the solution domain are inserted. Correspondingly, conditions are specified for the QX, QY, QZC and QZ indices where the nodal field variables are unknown. Since the solution obtained by the program depends intimately on the body of data, the program is queried to print out all data that have been input. This enables the programmer to check for input data errors. Statement 2970 marks the completion of these steps; the program is now ready to commence work on a particular fluid mechanics problem.

The loop to begin calculating the various element matrices starts with statement 3010. Once the element matrices $[TM\$]$ are computed for one element, they are assembled into the master or system stiffness matrix $[TM]$ by the code followed in statements 7740-7790. The non-linear terms appearing in the velocity and temperature expressions of the governing equations are formulated in statements 4110-5900. An iterative process compares each of these terms with a corresponding quantity in the linear symmetric $[TM\$]$ matrix until they converge in value. It is this comparison value that is then assembled with all similar values of the other elements to form the system $[TM]$ matrix, which now exhibits or reflects the non-linearity.

Since each element in the triangular mesh has six (6) nodes, the local node numbers are $I\$ = 1, 2, \dots, 6$. The global node numbers for the element are recovered from the parameter $NODE(K, I\$)$, which was read as input data for the element; that is, for element K , the node numbers $N(1)=NODE(K,1)$, $N(2)=NODE(K,2)$, etc. were introduced. Then the code in statements 7740-7790 loads the terms of the elemental matrices into their proper locations in the system matrices. Each time that a term of an element

matrix is placed in a location in the system matrix where another term has already been inserted, this new term is added to whatever value is there. A similar loading process takes place for the right-hand-side column vector in statements 7810-7940.

After all the elements have been processed in this fashion, the assembled system equations are ready to be modified to account for the boundary conditions or phenomena. This is done by statements 7980-8060. Thus, at the conclusion of statement 8060, the system equations possess the form

$$\begin{bmatrix} TM \end{bmatrix} \begin{Bmatrix} X \end{Bmatrix} = \begin{Bmatrix} RHS \end{Bmatrix}$$

$$\text{where } \begin{Bmatrix} RHS \end{Bmatrix} = \begin{Bmatrix} Kq \end{Bmatrix} = \begin{pmatrix} QX+gBTm \\ QY \\ QZC \\ QZ \end{pmatrix}, \text{ and } \begin{Bmatrix} x \end{Bmatrix} = \begin{pmatrix} u \\ v \\ P \\ T \end{pmatrix}$$

Not all of the components of the column vector $\begin{Bmatrix} RHS \end{Bmatrix}$ are known because the Q values at nodes where velocity, pressure or temperature is specified are unknown; that is, at each node i , either u_i , v_i , or T_i is known on the one side, or Kq_i is known on the other. A similar relationship exists at the corner nodes for pressure and QZC values. The only Q 's that can be specifically labeled as

heat fluxes are the QZ's, since they directly relate to temperature parameters within the system.

The only thing that remains to be done now is to call a compatible linear equation solver to produce the nodal variables sought. In this case, LEQT2F was chosen because of its speed and accuracy.

The following is a list of the symbols and descriptions utilized in coding the above program:

| <u>Symbol</u> | <u>Description</u> |
|---------------|---|
| NCASE | interger which specifies the type of problem to be solved: NCASE=1, 2-D plane problem NCASE=2, axisymmetric problem |
| NN | number of nodes in solution domain |
| NNCN | number of corner nodes |
| NE | number of elements |
| XC(I),YC(I) | global coordinates of node I |
| NODE(J,I) | J=1,2,...NE; I=1,2,...6 node numbers associated with element J |
| NVS(I) | node number where velocity or temperature is specified |
| NPS(I) | node number where pressure is specified |
| VELU | specified nodal u velocity |
| VELV | specified nodal v velocity |
| PNP | specified nodal pressure |

| <u>Symbol</u> | <u>Description</u> |
|-----------------|--|
| TNT | specified nodal temperature |
| NQS(I) | node number where a Q value is specified; QX and QY are specified only at internal nodes, while QZC and QZ may be specified at either external or internal nodes |
| QXNS | specified nodal value of QX |
| QYNS | specified nodal value of QY |
| QZCNS | specified nodal value of QZC |
| QZNS | specified nodal value of heat flux QZ |
| XC\$(I),YC\$(I) | local coordinates of node I |
| TM\$ | element stiffness matrix |
| TM | system stiffness matrix |
| DEL | area of a triangular element |

The program output begins with a statement declaring the type of problem to be solved - either nonlinear two-dimensional or axisymmetrical. Next, all input data are printed and labeled for easy identification. To ensure the validity of the solution, the printed input data should be carefully checked against the intended input. A statement following these printed data identifies which nodal parameters are associated with which system nodes (re-membering specifications of the finite element analysis called for a value of both velocities along with a temperature

at each node, while a pressure value could be defined only at the corner nodes). The complete continuum solution follows in the form of a numbered list, in which the integer appearing at the far left of this list designates the node number, or multiple of it in cases above $I=35$, and the figure on the right representing the value of the nodal variable in double precision.

D. NUMERICAL RESULTS

Complete numerical listings of the field variables for both the Couette flow problem and the steady state heat transfer problem are shown in the two computer program outputs.

The velocity profile for the linear Couette flow, i.e. node numbers 1-5, 6-10, etc., revealed that the finite-element method of analysis agreed with the exact solution of this shear-type flow out to the sixth decimal place. This is evident by the fact that all five of the FEM points lie exactly on the smooth curve depicting the exact solution in Figure 15.

The approximate steady state isotherms of Figure 16 are directly related to the nonlinear temperatures (node numbers 83-117 of the second set of nodal variables) in

the heat transfer problem. These isotherms, or constant nondimensional temperature lines, vary in value from +1.0 on the hot temperature wall, to -1.0 on the cold temperature wall. The equation used for deriving these values at all thirty-five nodal points within the solution domain was

$$\theta = \frac{T - T_M}{T_H - T_M} \quad (4.29)$$

where T is the nodal temperature and T_M is the mean temperature of the fluid defined at $T_M = \frac{(T_H + T_C)}{2}$.

The general shape and relative location of the various isotherms within the rectangular enclosure are somewhat similar to those of comparable heat transfer flows involving different Prandtl numbers, Grashof numbers and L/D ratios. However, due to the relatively low Grashof number of the present system (946.4), there is a total lack of a plateau in the center region of Figure 16 and the closely packed boundary layer flow near the walls is also missing. This boundary layer type flow is characteristic of much higher Grashof numbers such as those found in the comparative examples in [3], [7] and [15] where the Gr_L ranged from 5000 up to 18000.

Based solely on the thirty-five nodal point temperatures available from the solution, the isotherms were sketched as

shown in Figure 16. Lacking additional information, the shape of the contour lines between such nodes were linearly approximated, to a large extent, without speculating as to their exact curvature.

The actual height and width of the enclosure was normalized to y^* and x^* , respectively, for easier interpretation of the figure.

V. EXPERIMENTAL PROCEDURE

A. ARRANGEMENT OF TEST APPARATUS

The experimental apparatus was arranged so as to allow the study of an essentially two-dimensional fluid flow.

The major components of the apparatus consisted of the test platform, which housed the rectangular enclosure (Figure 6), a control system made up of two water circulators that maintained the vertical copper walls of the test platform at desired temperatures (Figure 7), and a large (250 mm DIA) plano-convex glass lens for reducing the object (8.5 x 1.875 inch vertical rectangular cavity) down to a smaller image size that could be completely captured on the 4x5 inch holographic plate (Figure 11).

The rectangular enclosure holding the fluid under investigation was 8.5 inches high, 7 inches long, and 1.875 inches wide. This test cavity was sandwiched between two plexiglas water reservoirs providing constant circulation by means of manifold connections on their tops and bottoms. Hot and cold water drains were located on top of the left and right reservoirs, respectively. Similarly, on the bottom were the hot and cold water inputs. The

inner walls of these two reservoirs were formed by quarter inch thick oxygen-free copper plates. Also, these same copper plates comprised the principle walls of the rectangular enclosure, with the "side walls" being made of plate glass, in order to allow visual observations. Six thermocouples were attached to each copper wall and then connected to a multichannel recorder for temperature monitoring purposes. The time needed for each copper plate to reach its respective equilibrium temperature once the water circulators had been turned on was approximately 39.8 seconds. For comparison purposes, it took the system just under one hour (58.1 minutes) for the 50-HB-3520 lubricating fluid to attain a steady equilibrium temperature under the same experimental conditions.

An important constraint imposed by interferometry is that the total distance traveled by the object beam must be nearly identical with the total path length of the reference beam, if the index of refraction throughout is uniform. Since the fluid in the rectangular enclosure possessed a refraction index of 1.461 and laser light along the object beam had to travel through 7 inches of this fluid, then the corrected path length through the test cavity was 10.23 inches, or a net increase of over 3 inches.

With this in mind, the equipment was arranged in a semi-elliptical pattern on a heavy table supported at six critical areas by inflated inner tubes. These were to act as stabilizing devices. Equipment could not be arranged on an exact ellipse due to the fact that a distance of four feet, five inches alone was needed from the object to the plano-convex lens out of a total table length of eight feet. Even so, the turning mirrors were located on the apexes of the "shortened" minor axis, the beam splitter at one end of the major axis, and the aqueous hologram holder at the opposite end (Figure 8). Spatial filters were employed to clean up each beam and expand it. A diverging lens was inserted just after the object beam spatial filter in order to expand this beam to proper size before it reached the rectangular test slit. Also, a collimating lens was placed between the spatial filter and hologram holder along the reference beam. Finally, a large diffusing screen made from a piece of developed film mounted on plexiglas and secured in a rigid metal frame, plus the test platform itself, were placed between the object mirror and the hologram holder (Figures 9 and 10).

Choosing the correct hologram holder is very important in live fringe holography. The reconstructed virtual image

must exactly match the original object. Unfortunately, after the processing of a hologram, the emulsion on its surface tends to dry, thus causing an unwanted displacement of the virtual image. One procedure that may be used to circumvent this problem is to choose a holder that maintains the hologram in aqueous surroundings, such as was utilized in this experiment. Also, in order to keep the hologram perfectly rigid during and after processing, the exposed glass plate was secured in a removable metal frame complete with handle. In this way, exact replacement in the hologram holder after processing posed no problem. Two micrometers built into the holder's top and left side were then used for fine adjustment of the hologram.

The test cavity or rectangular enclosure was filled with a very highly viscous fluid (actually a lubricant) produced by Union Carbide and known as "UCON" 50-HB-3520. This fluid was required to possess physical properties such that Rayleigh numbers, based on cavity width, of the order of 10^4 - 10^5 could be obtained in the apparatus, at accurately measurable temperature differences. Since $(T_H - T_C)$ was held constant throughout the experiment, only one Rayleigh number was calculated. Its value of 1.0755×10^4 was well within the above tolerance zone. Worth mentioning

is the generally accepted prediction that above $Ra \approx 2.0 \times 10^4$, the phenomena known as "secondary flows" begin to occur. Obviously, such was not the case in this experiment.

A scribed grid pattern was attached to the back side of the test cavity to assist in alignment of the fringes. Two water circulators were connected by tubes to manifold nipples on the reservoir ends of the test platform. One circulator was set to deliver distilled water at 20°C (cold temp.) and the other at 25°C (hot temp). By using slide valves, the amount of water expended from the circulators could be regulated and controlled.

The experimental procedure was initiated only after the entire system had been carefully aligned. The rectangular enclosure was allowed to sit undisturbed for a period of at least several hours to ensure an equilibrium temperature state throughout the fluid. Then, a shutter was placed directly in front of the beam of a 3 milliwatt, helium-neon continuous wave laser serving as the coherent light source. After an Agfa-Gevaert Inc. 10E75 holographic recording plate was placed in the holder, the shutter was opened and the plate was exposed for one and one-half seconds (Figure 13). This plate was then removed, developed, and replaced in its exact position. Both circulators were

then turned on, and a continuous flow of water at 20°C and 25°C was allowed to cycle through the reservoirs on the test platform. Once fringe lines appear, their visibility can be strengthened by following the procedure outlined in the next subsection.

If one word could be used to describe the single most important factor determining the success or failure of this experiment, it would have to be rigidity. All relatively light-weight gear, such as; the laser, turning mirrors, spatial filters, and the plano-convex lens had to be weighted down to make them immovable. The test platform, in which the rectangular cavity was located, was sufficiently heavy on its own to preclude it from having to be additionally weighted down. The hologram holder already came with a very heavy base attached. All connecting devices, including the tubes transporting heated water to the plexiglas reservoirs and plastic sleeves housing the thermocouple leads were securely taped together to prevent vibration or motion. Any such random vibration would cause the fringe patterns to become lost.

The viewing of these fringes and the subsequent collecting of data can be accomplished by positioning the appropriate camera in a direct line with the rectangular

enclosure, plano-convex lens, and hologram holder (Figure 11). A television monitor (Figure 12) was employed for convenient viewing of the fringe patterns in an area adjacent to the experimental set-up.

B. HOLOGRAPHIC INTERFEROMETRY APPLICATIONS

Holographic interferometry is an excellent technique for developing interference fringe patterns, which may in turn be evaluated to quantitatively provide an accurate temperature field throughout the domain of the system.

Heat convection in a rectangular cavity would be difficult to analyze empirically. However, by replacing the sensors that would ordinarily be used to record temperature changes and flow rates with holographic technology, one can analyze directly the variation of the density fields within the rectangular enclosure. This technique also eliminates the inherent change in temperature and flow pattern caused by the physical insertion of the sensors into the test fluid.

Real-time holographic interferometry allows a continuous flow of information to be recorded at the precise time any changes in the observed fluid occur. Single exposure holograms are utilized with real-time interferometry. A time

sequence can be derived for each different viewing position, with the use of a single developed hologram.

Such an exposure technique consists of recording phase and amplitude information from an object, in this case the rectangular fluid enclosure, onto a holographic plate. The recording is accomplished through the use of a reference and an object (scene) beam, originating from a single source (Figure 13). After processing, the hologram is accurately repositioned in its holder. Illuminating the plate with the original reference beam results in the primary (virtual) image being projected onto the same area as was the object (Figure 14). By focusing the object and virtual image beams onto a film or focal plane, and then adjusting the system so that the two interfering wavefronts (object wave and reconstructed wave) coincide, fringes can be produced.

The hologram now can be finely adjusted to orient the fringes in either a vertical or horizontal reference frame. Likewise at this time, the fringe patterns can be made to appear more visible by varying the beamsplitter to increase the intensity of the reference beam while decreasing that of the scene beam. If the original object is changed or altered in any fashion by the effect of temperature, motion, or pressure, an exact superposition will create a reinforcement

or cancellation of the intensities of the two waves with the result being the establishment of a fringe pattern. A dark fringe is produced whenever the difference between the object and reconstructed wavefront involves an odd factor of $\pi/2$. Bright fringes occur when this difference equals an integer value of 2π .

By inserting a camera in-line with the scene (object) beam, but on the back side of the holographic plate, one can observe and record live fringe data. This technique provides a real time analysis of an unsteady system without the need for expensive and time consuming sensors and calibration.

After processing has been completed, problems arising from live fringe single exposure holography include; displacement of the virtual image due to drying emulsions on the holographic plate, and non-exact replacement of the hologram in its holder. If any relative motion whatsoever has transpired between pieces of the experimental equipment during or after replacement of the hologram, the fringe patterns may be destroyed.

It was this last problem that caused a particularly detrimental effect on the experimental results of this thesis. Somewhere within the system (apparatus arrangement)

there existed a source of motion or a piece of gear slightly off the horizontal reference plane that completely eliminated these fringe formations almost immediately after they evolved. The exact source(s) was never totally isolated, but the possible choices were reduced down to two, the water circulators and/or the support stand of the hologram holder.

VI. CONCLUSIONS

The finite-element method was incorporated into steady state and time dependent computer programs for analyzing laminar convective heat transfer between parallel plates. Two sample cases were tested utilizing the general steady state program. In each case, values of derived field variables compared favorably to either an exact solution, in the case of the Couette flow problem (Figure 15); or to similar theoretical results, in the case of the heat transfer problem (Figure 16). The exact solution of the velocity profile for Couette flow was obtained by programming the analytical expression given by equation 5.5 in [11].

After successful steady state results were achieved, a second computer program was then designed to take into account the previously-neglected transient behavior of the system. A major portion of the steady state fluid mechanics problem was interfaced with a series of subroutines, wherein the time-dependent terms were to be calculated, to yield a total solution to the governing system of equations and associated boundary conditions. Time itself became a limiting factor in the completion of this second program.

A problem associated with the convergence of the field variables remains to be resolved.

In the experimental phase of this thesis, five attempts were made to produce live fringe formations through the use of holographic interferometry. In only one of these attempts was there observed a momentary interference pattern, corresponding to the temperature gradient, across the test section. This observation lasted approximately three (3) seconds after the water heaters/circulators were activated.

The main factor(s) influencing this inability to acquire such live fringes, on film, was the necessary exclusion of the hologram holder from the recording plane because of limited table length and/or the vibrations generated by the water circulators used in the experiment. Either of these detrimental conditions could have eliminated completely the formation of interference fringe patterns.

Due to considerable time delay in the acquisition of some of the experimental apparatus, no further documentation of real time holographic interferometry study could be made beyond the previously-mentioned five attempts.

APPENDIX A

FIGURES

Figure 1
Rectangular Enclosure

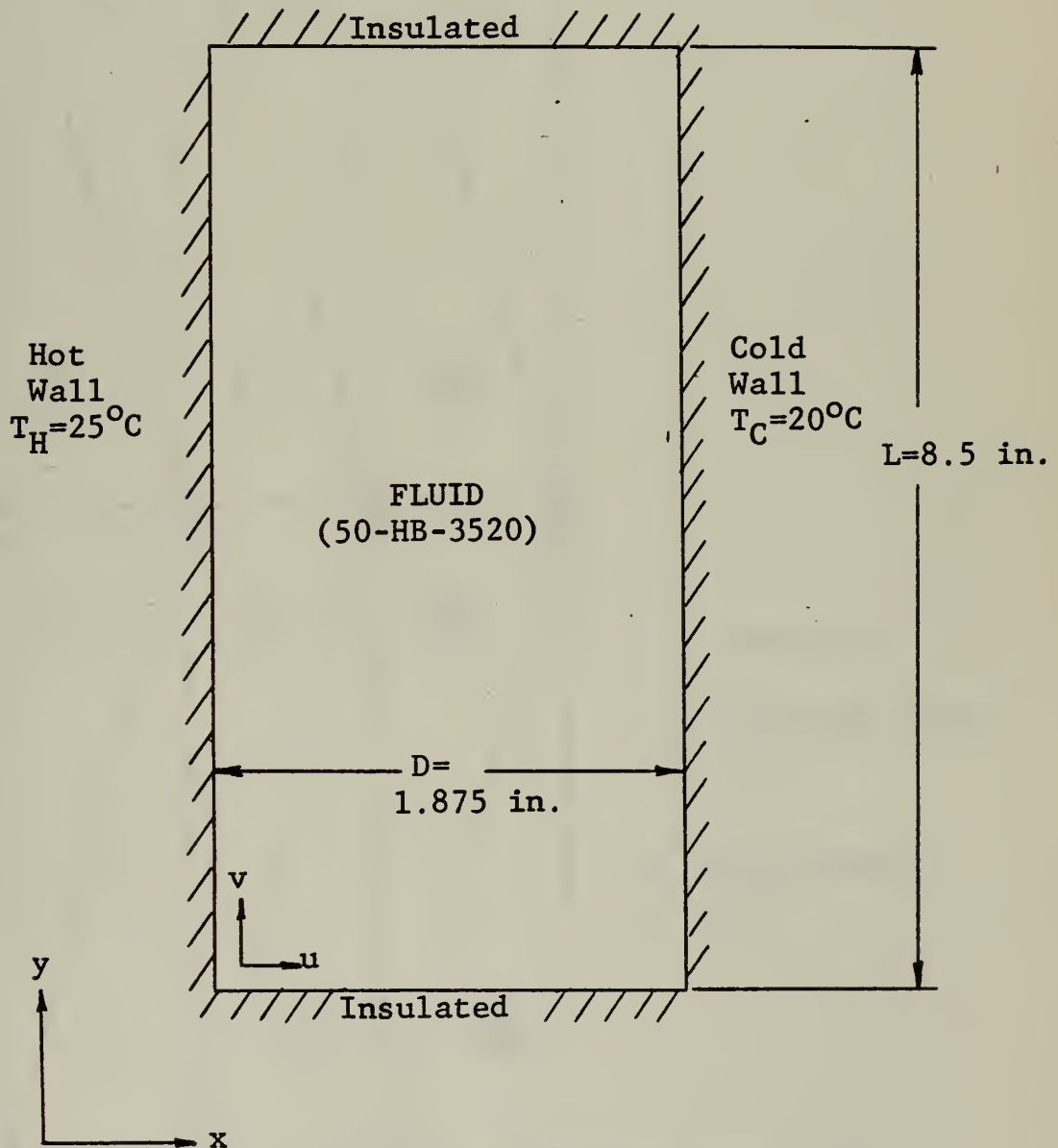


Figure 2
Discretization of Solution Domain

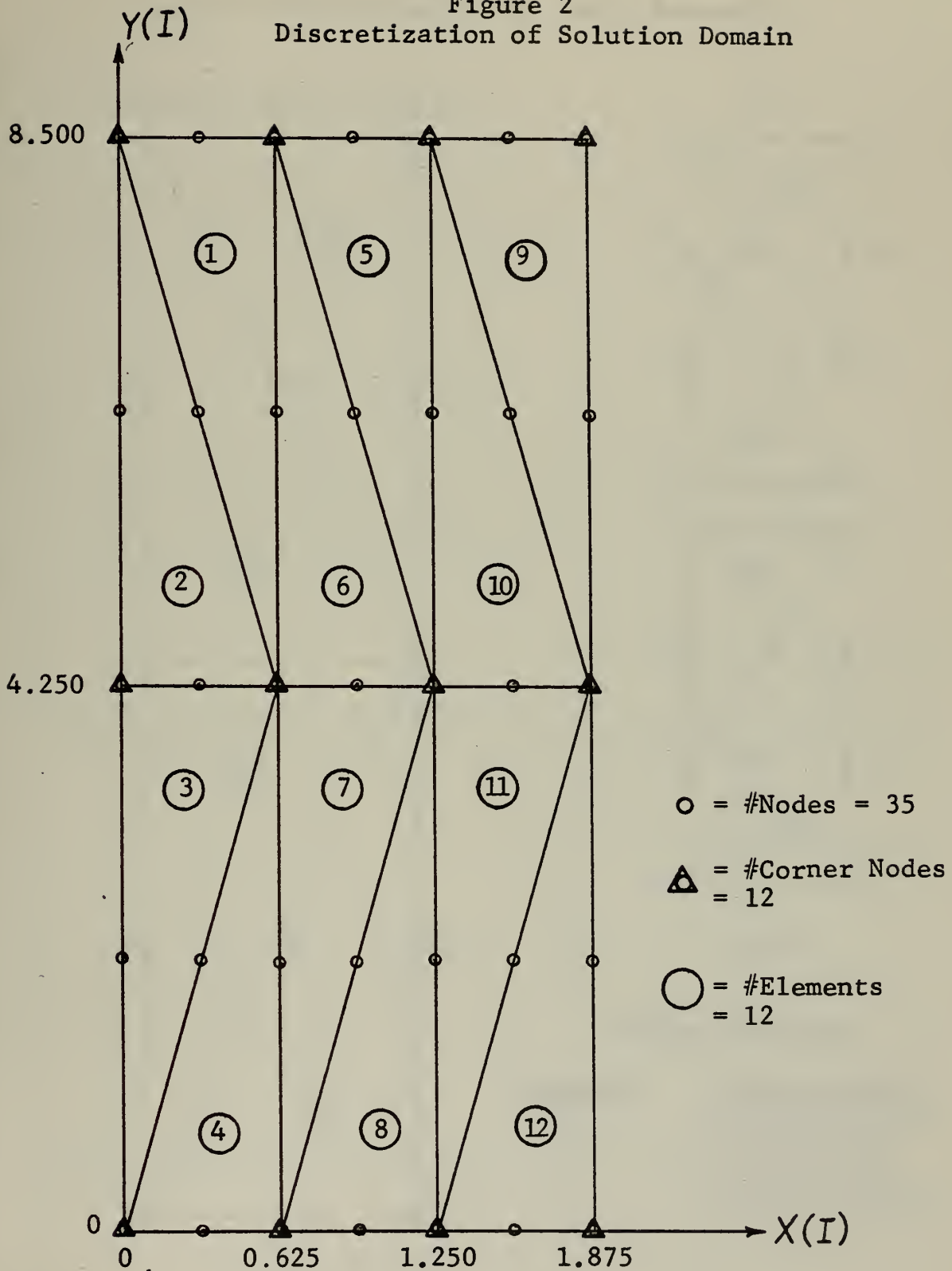
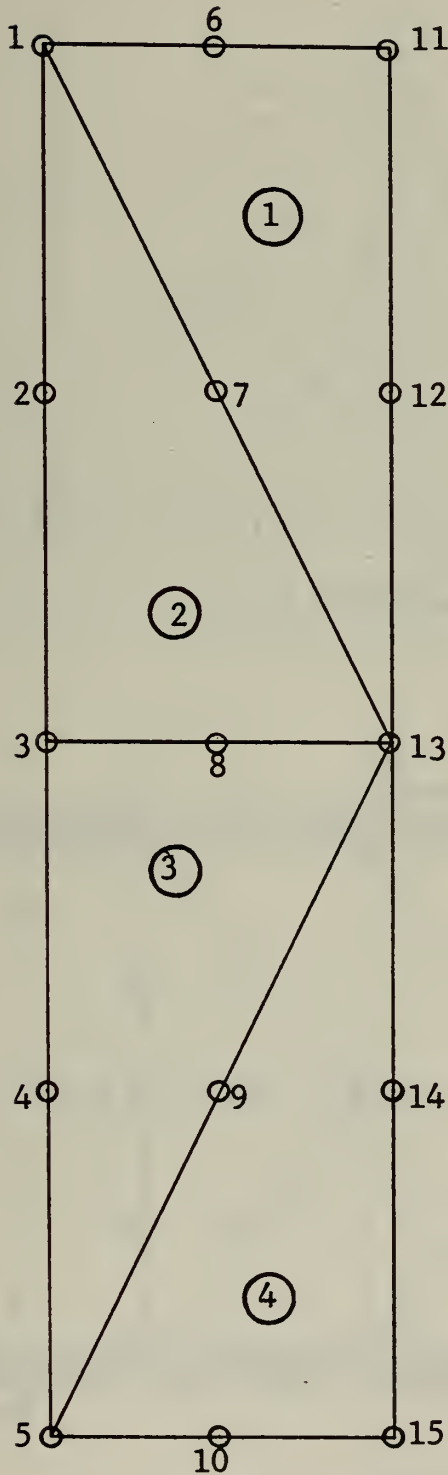
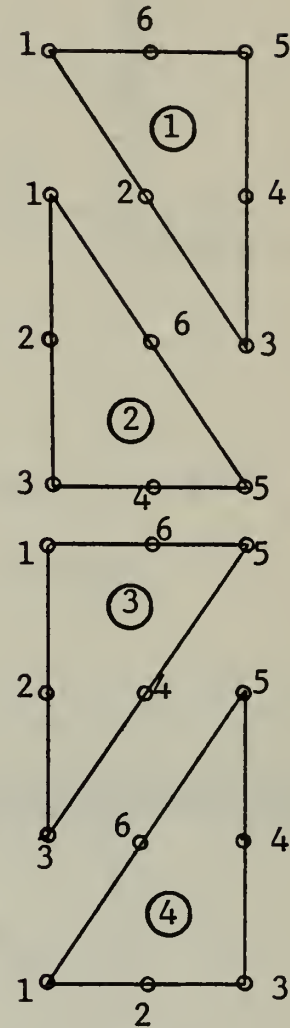


Figure 3
Global Node-Numbering vs. Local Numbering



Global



Local

System Topology

| <u>Element#</u> | <u>Global Node#'s</u> |
|-----------------|-----------------------|
| 1 | 1, 7, 13, 12, 11, 6 |
| 2 | 1, 2, 3, 8, 13, 7 |
| 3 | 3, 4, 5, 9, 13, 8 |
| 4 | 5, 10, 15, 14, 13, 9 |

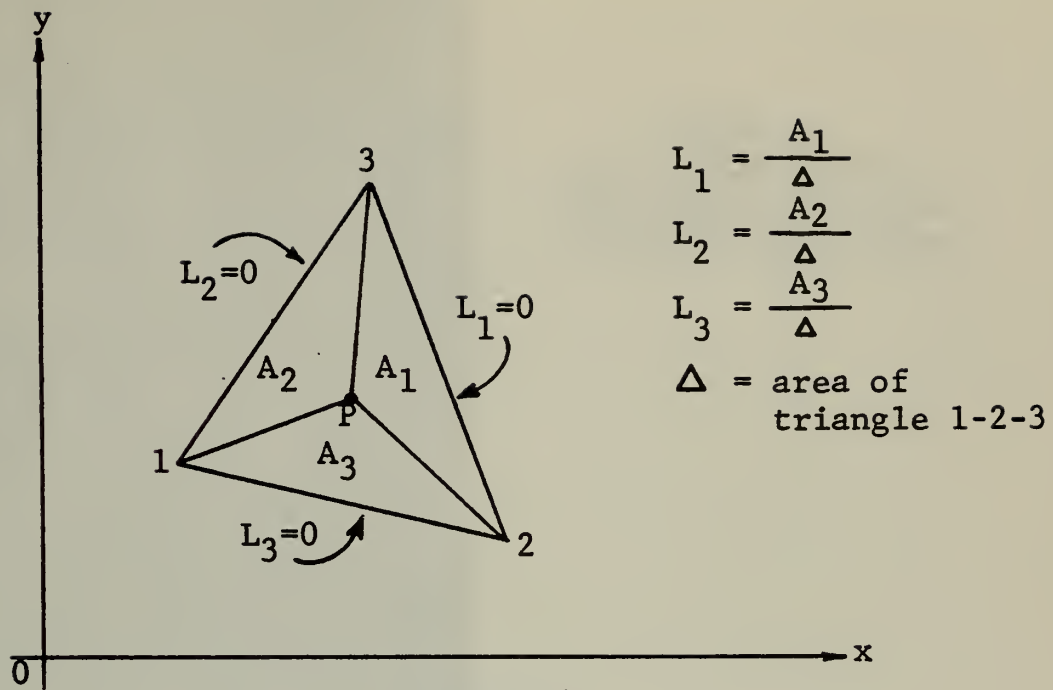


Figure 4
Area Coordinates for a Triangle

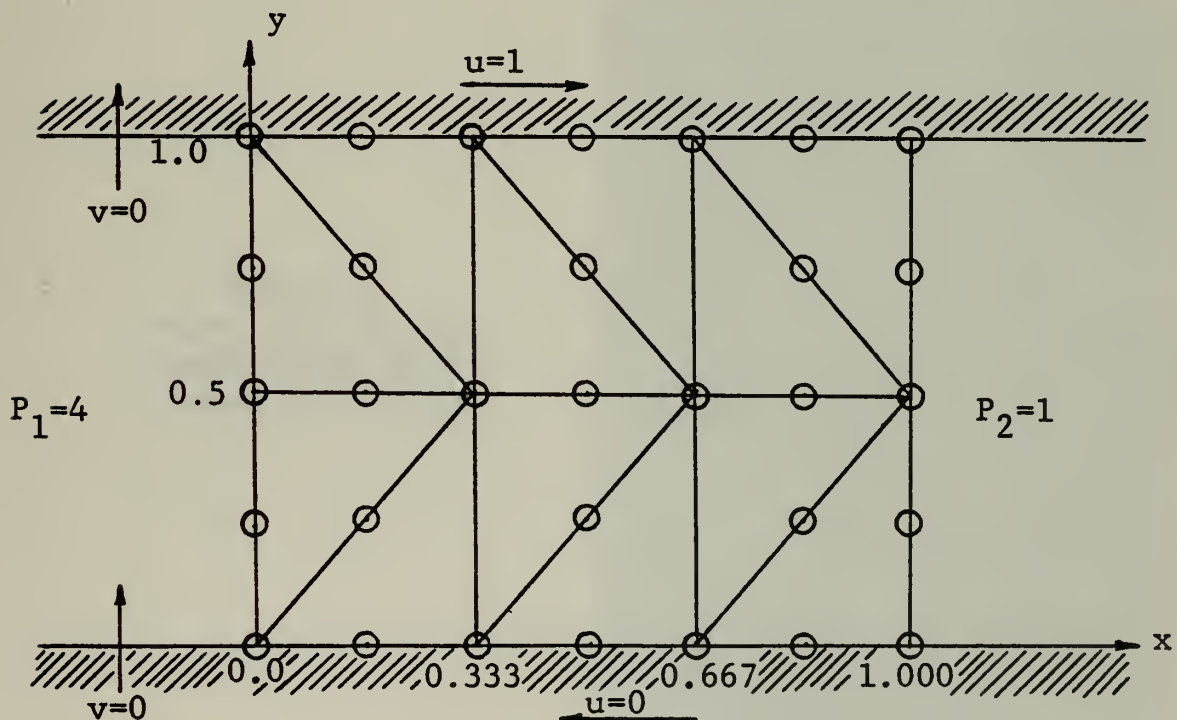


Figure 5
F.E.M. Analysis of Couette Flow

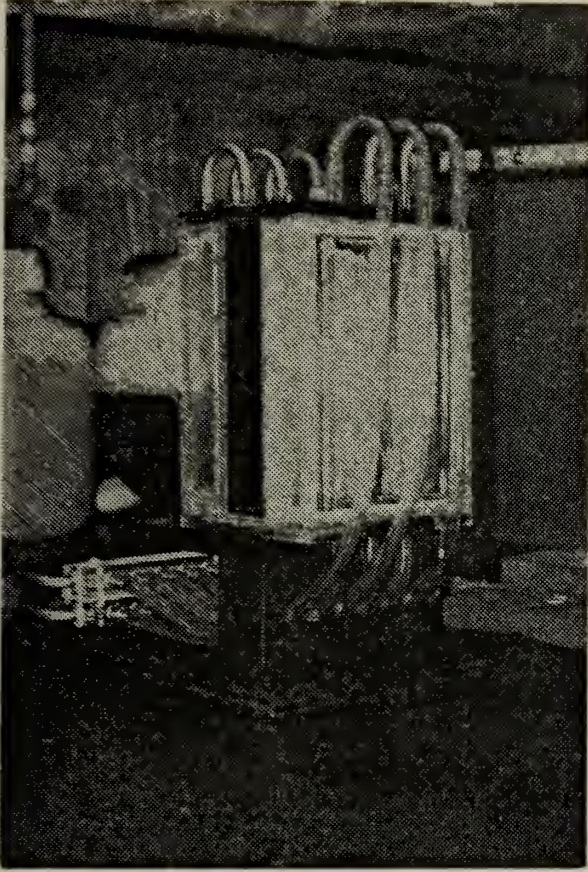


Figure 6
Test Platform with
Rectangular Enclosure
and Water Reservoirs

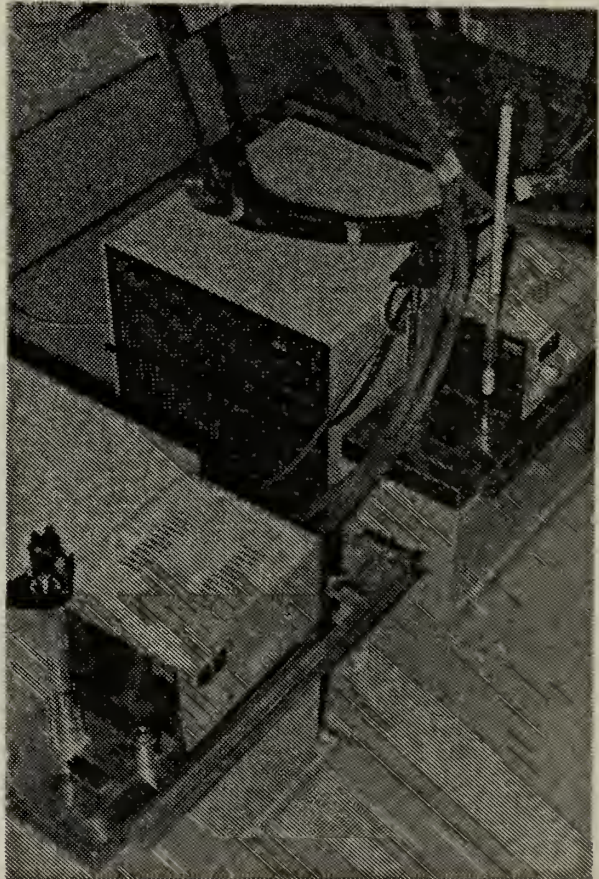
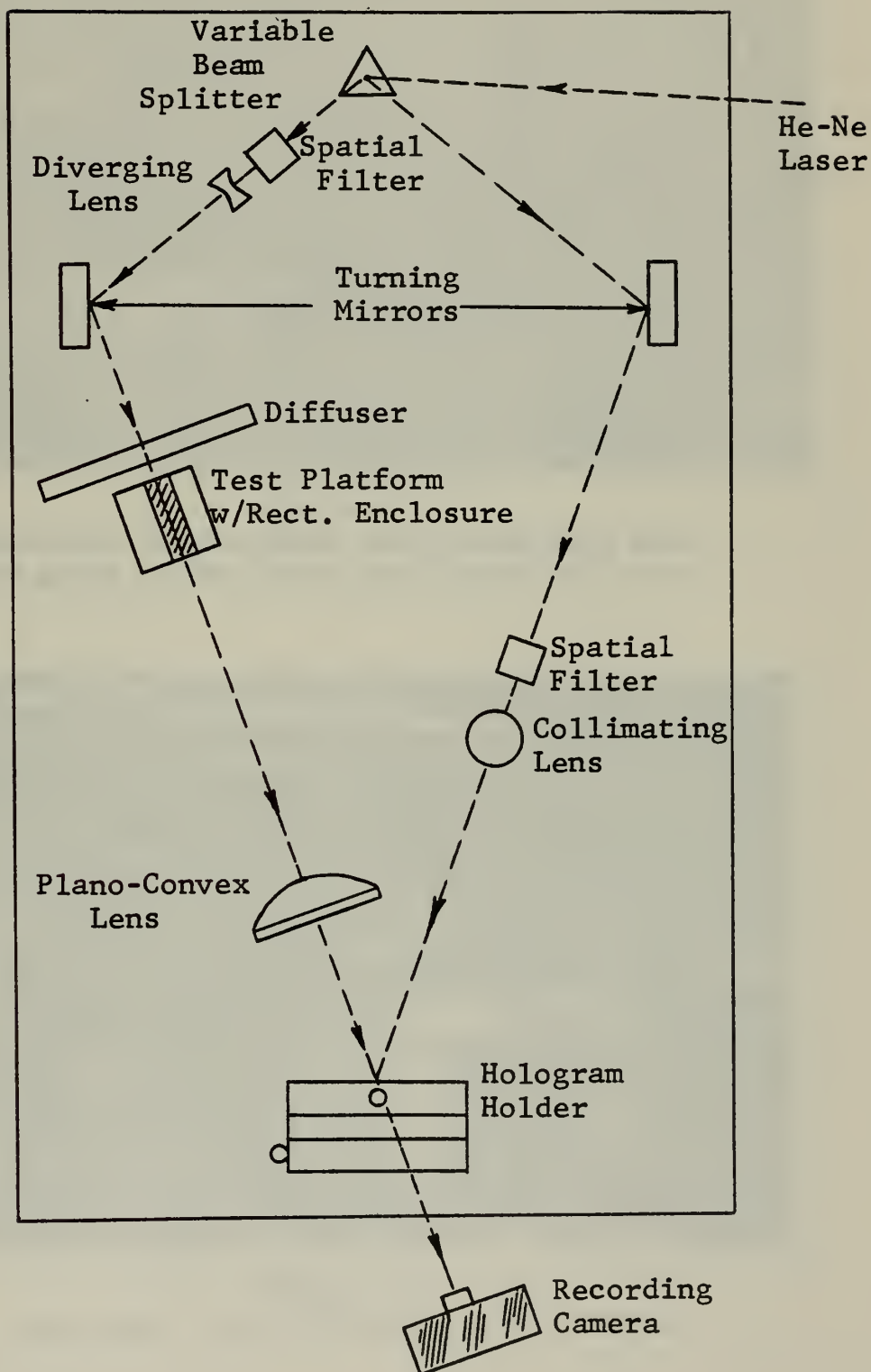


Figure 7
Water Heaters and
Circulators with
Connecting Tubes

Figure 8
Table Arrangement (top view)



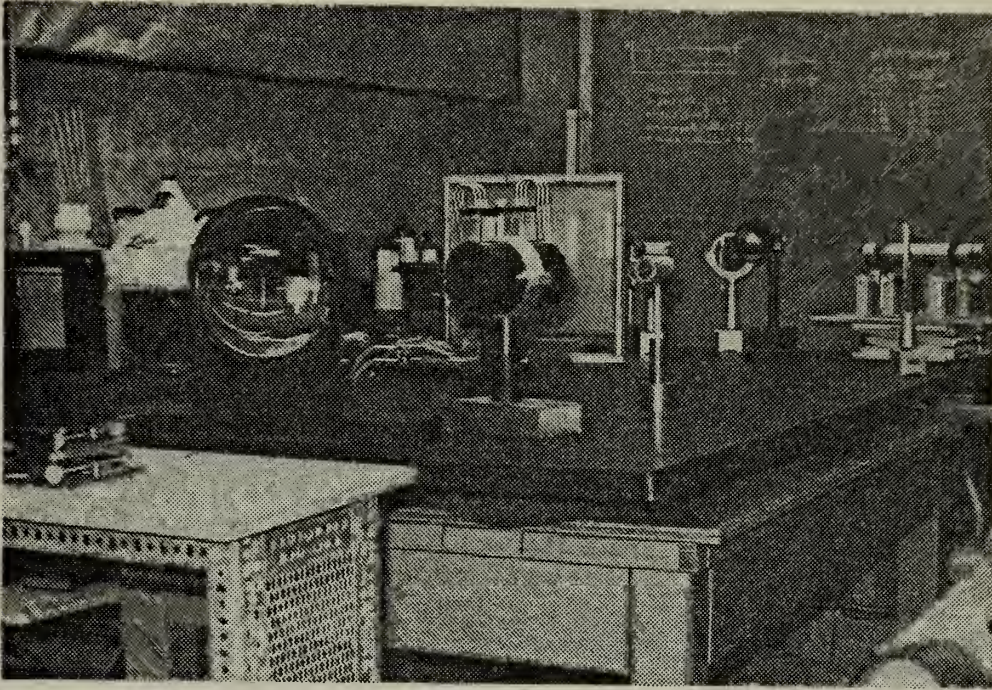


Figure 9
Apparatus Arrangement with Reference Beam
Oriented on the Right-hand-side of Table

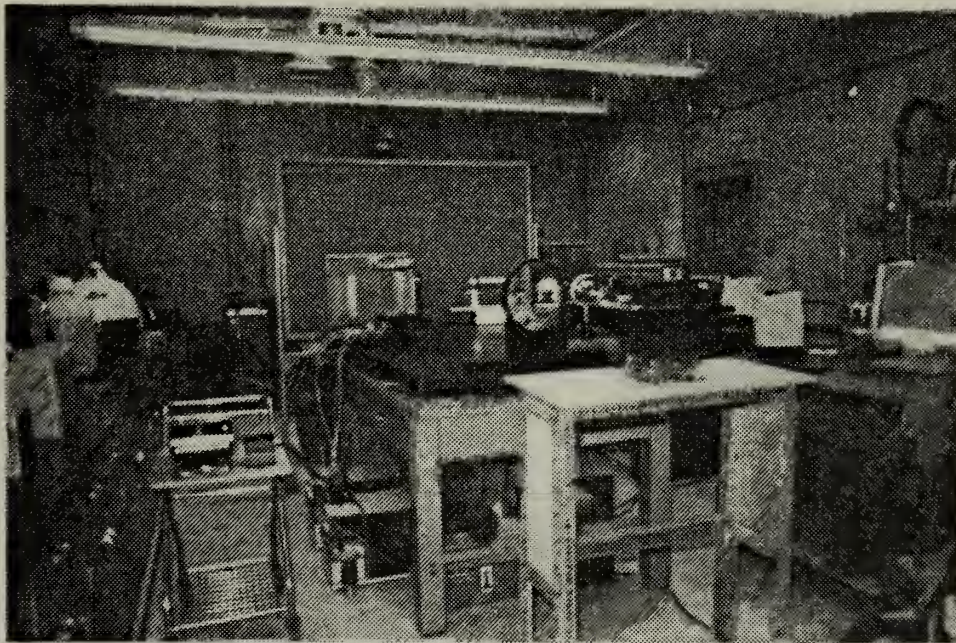


Figure 10
Panoramic View of Experimental Layout

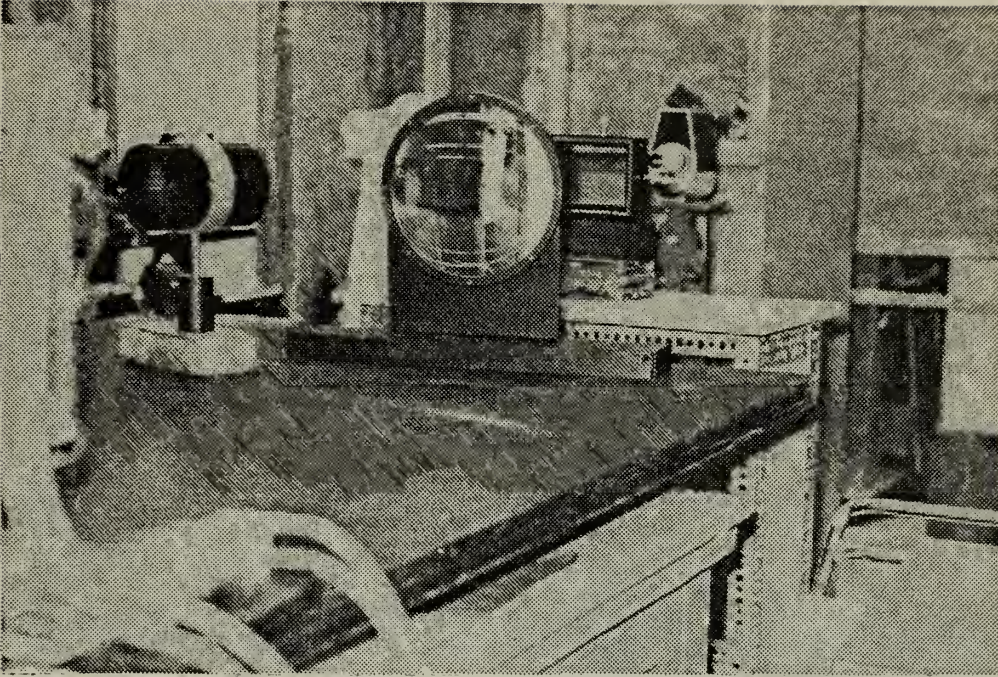


Figure 11
Direct Line-up on Object Beam from
Test Platform to Recording Camera

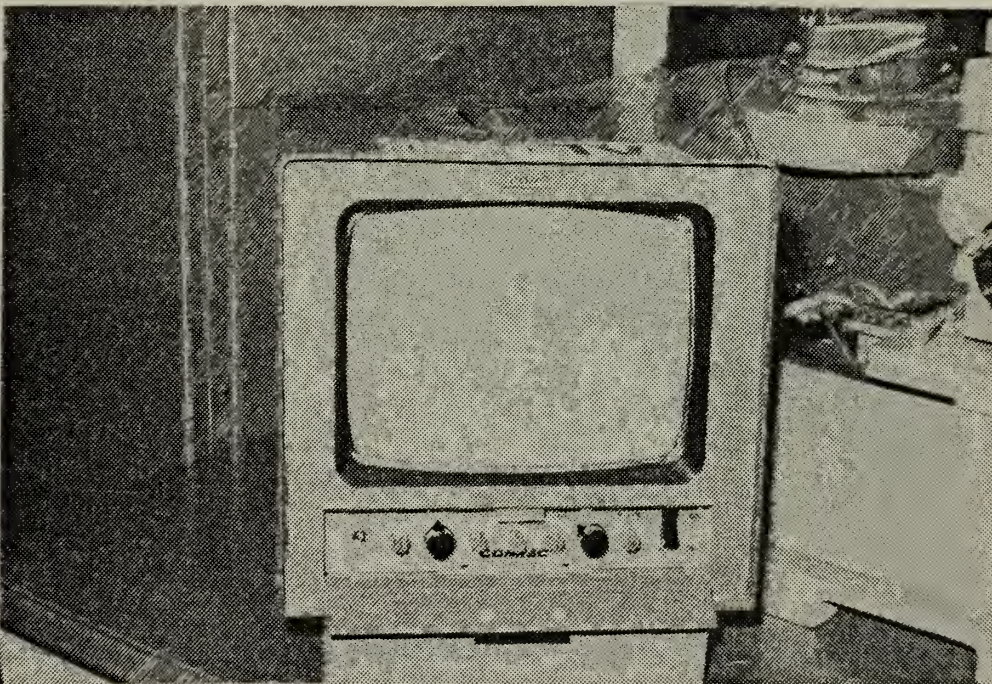


Figure 12
Television Monitor Used for Convenient Viewing of Fringes

Figure 13
Holographic Recording (top view)

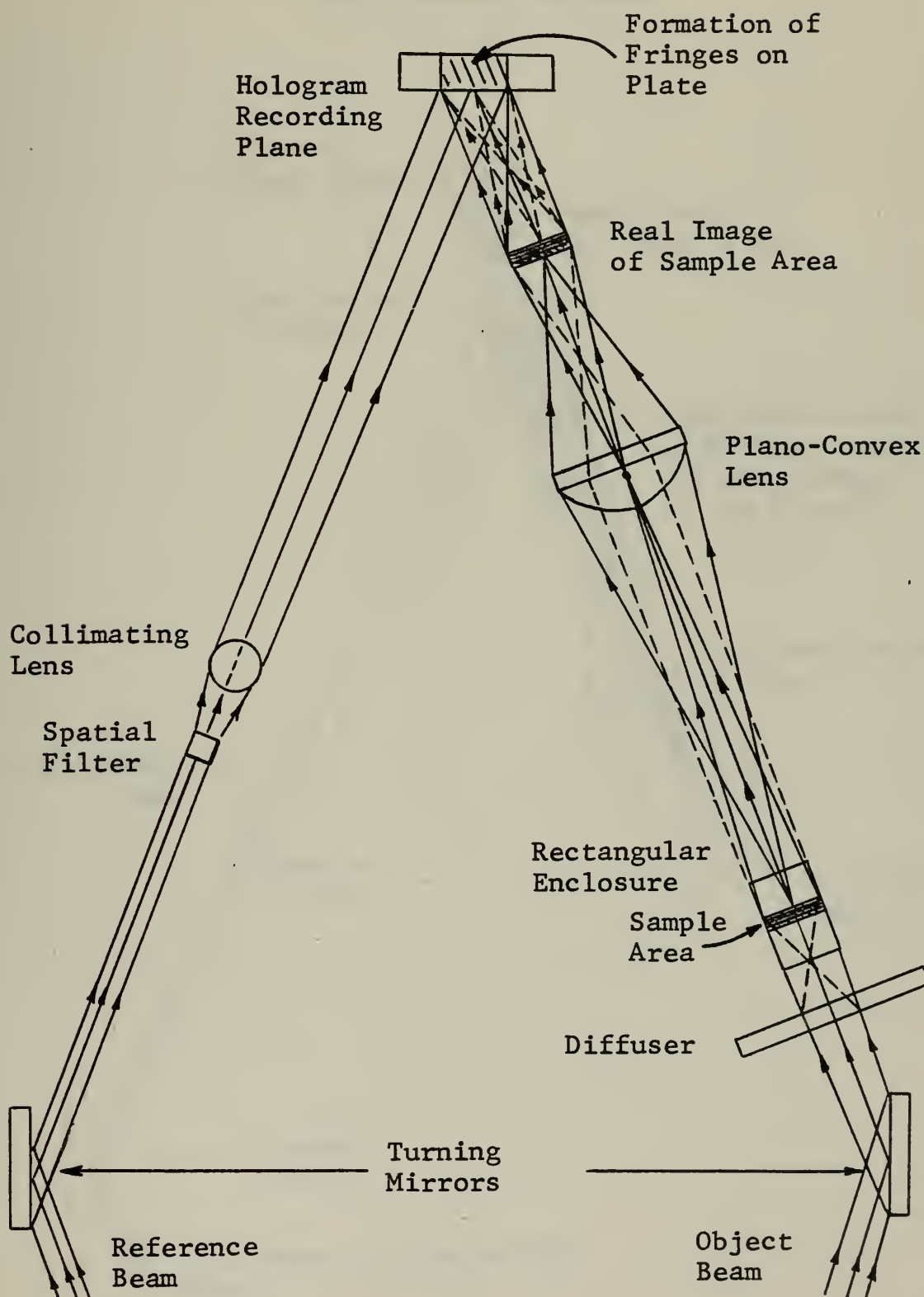
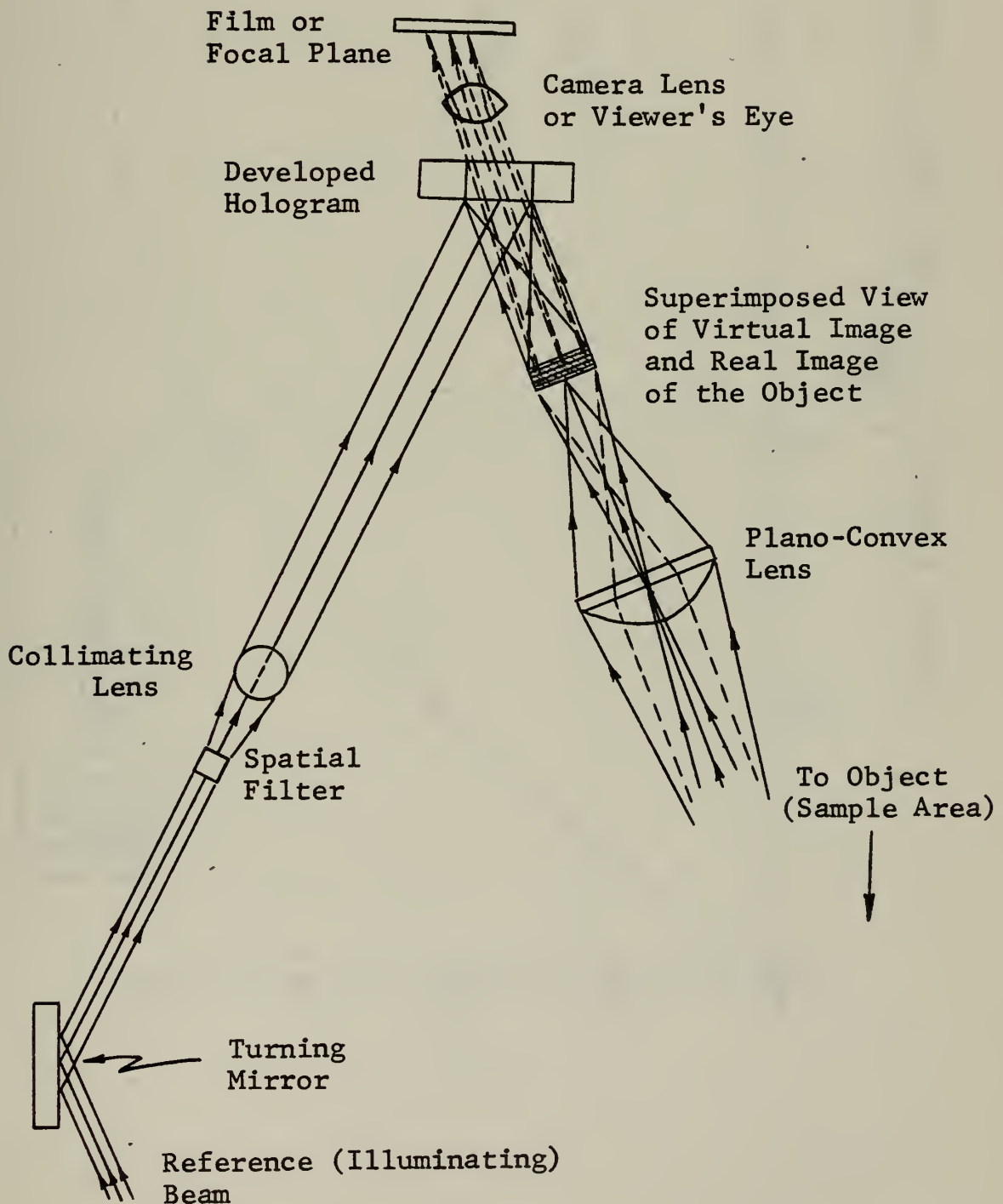


Figure 14
Reconstruction and Recording of
Interference Patterns



U VS. Y
FEM & EXACT SOLUTIONS
 $DP/DX = -3.0$

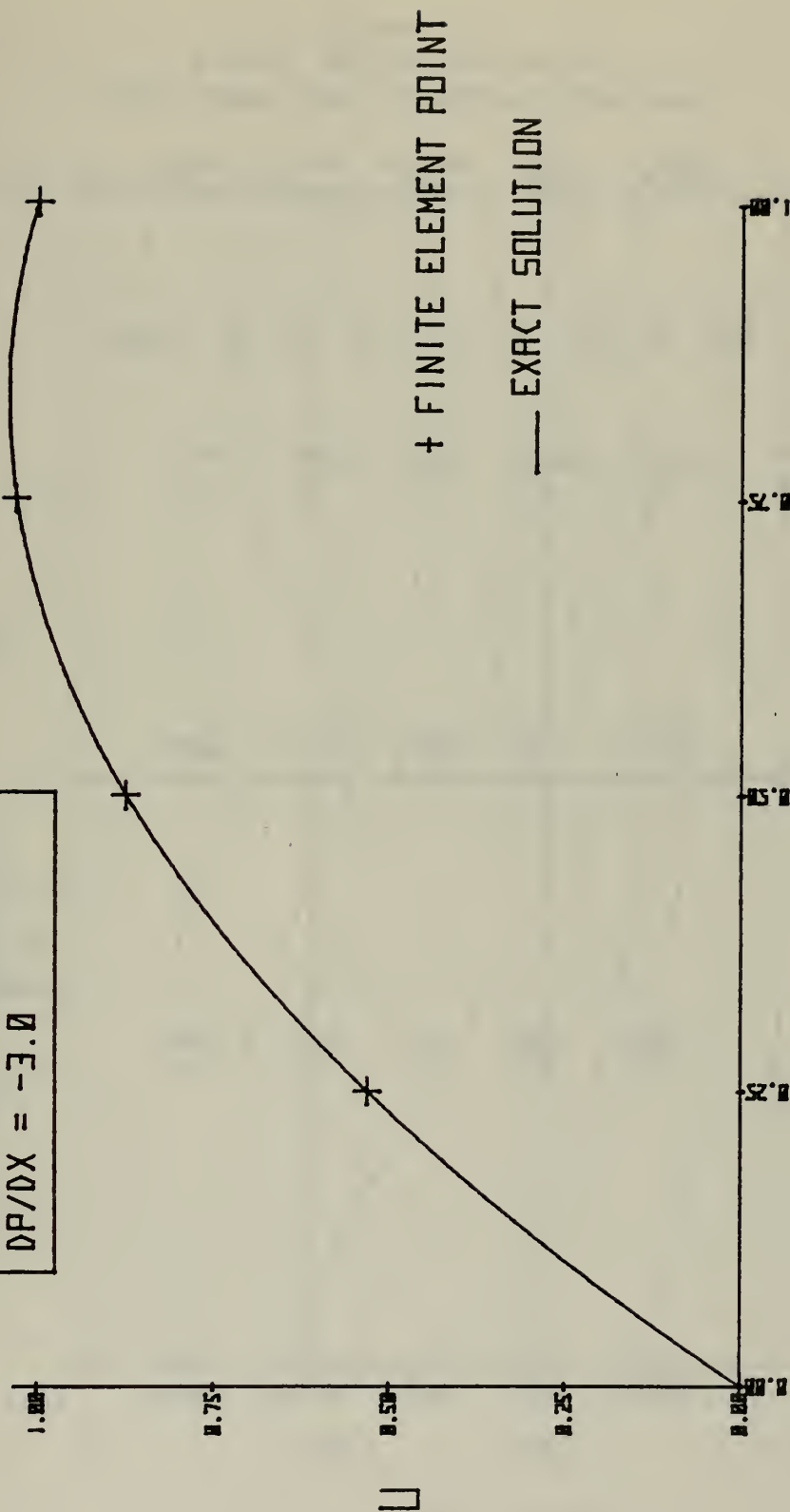
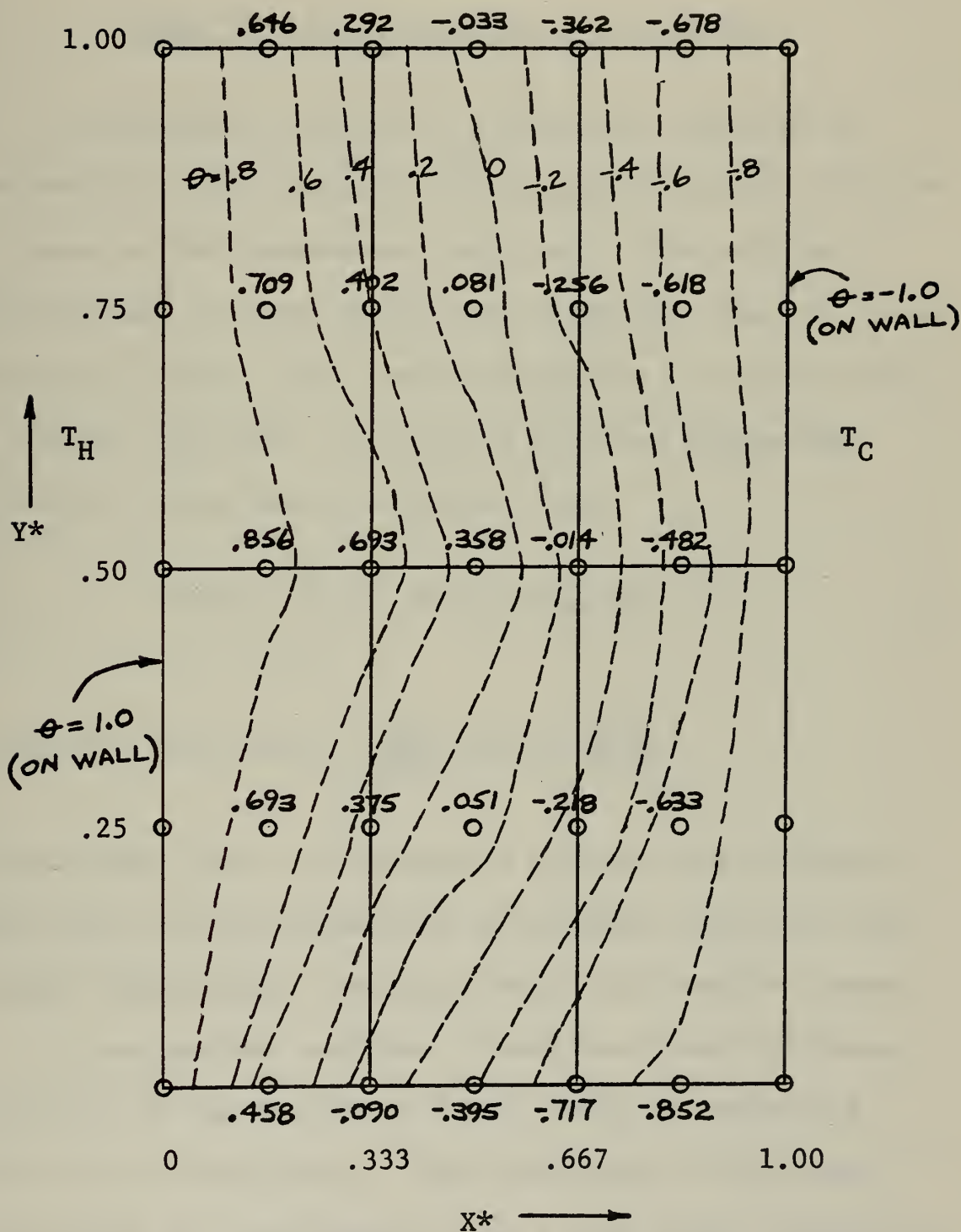


FIGURE 15
VELOCITY PROFILE FOR LINEAR COUETTE FLOW

Figure 16
Steady State Isotherms
(Nonlinear Heat Transfer Problem)



$Gr_L = 9.464 \times 10^2,$

$Pr = 1.0755 \times 10^4,$

$L/D = 4.53$

APPENDIX B

BRIEF REVIEW ON CALCULUS OF VARIATIONS

A fundamental problem in differential calculus is extremizing (maximizing or minimizing) a function $f(x)$ for a range of the independent variable x . The problem in variational calculus is also extremization; however, it is concerned chiefly with the extremization of a functional. A simple functional, in terms of only one independent variable, would have the typical form

$$I(\phi) = \int_{x_1}^{x_2} F(x, \phi, \phi_x, \phi_{xx}) dx$$

where $\phi = \phi(x)$ and $\phi_x = \frac{\partial \phi}{\partial x}$, $\phi_{xx} = \frac{\partial^2 \phi}{\partial x^2}$.

Summarizing, the two branches of calculus are related in that both are concerned with an extremum; one deals with number spaces while the other deals with function spaces.

In variational problems a functional which is characteristic of the problem is first formed in terms of a function (or functions). Then variations of this same functional are investigated with a view toward extremizing the functional. In some cases this approach results in a

closed form, exact solution. But more often, the problem must be solved by an approximate method. One such method is the Rayleigh-Ritz technique. This approach is preferable to the direct application of finite difference methodology to solve the differential equation with its associated boundary conditions, because the functional can often be used to assure convergence of the approximate solution.

A simple example of variational calculus is the problem of finding the plane curve joining two points (x_1, y_1) and (x_2, y_2) which has the shortest length. The solution sought here is the function $y(x)$ describing the curve of shortest length; the corresponding functional is the length of the curve given by

$$I(y) = \int_{x_1}^{x_2} \sqrt{1 + \left(\frac{dy}{dx}\right)^2} dx$$

Using the method of variation of calculus implies that of all the curves

$$Y(x) = y(x) + \epsilon \delta(x)$$

which pass through the given end points, the shortest one $y(x)$ must be selected. The problem thus reduces to finding the function $y(x)$ that makes the integral $I(y)$ a minimum.

Generally, in order to minimize the integral

$$I(y) = \int_{x_1}^{x_2} F(x, y, y') dx$$

where $y' = \frac{dy}{dx}$, the function $y(x)$ must satisfy the boundary conditions and the Euler-Lagrange differential equation

$$\frac{\partial F}{\partial y} - \frac{d}{dx} \left(\frac{\partial F}{\partial y'} \right) = 0$$

The previous result could be extended to several dependent and independent variables. For example, in order to minimize the integral

$$I(\phi) = \iint_A F(x, y, \phi, \phi_x, \phi_y) dx dy$$

in which ϕ_x and ϕ_y are the partial derivatives of ϕ with respect to x and y , respectively, the general function ϕ must satisfy the Euler-Lagrange differential equation

$$\frac{\partial F}{\partial \phi} - \frac{\partial}{\partial x} \left(\frac{\partial F}{\partial \phi_x} \right) - \frac{\partial}{\partial y} \left(\frac{\partial F}{\partial \phi_y} \right) = 0$$

in addition to the specified boundary conditions.

In the past two decades, since the advent of high speed digital computers, the variational formulation has been quite extensively employed in the fields of structural

and continuum mechanics. Important variational principles such as least work, minimum strain energy, minimum potential energy, and Reissner's variational theorem of elasticity have been well developed and are documented in standard textbooks. However, similar variational principles applicable to fluid mechanic problems have not been as comprehensively developed. Calculus of variations has, until only recently, been utilized sparingly in the field of fluid mechanics.

THIS IS A 2-D NONLINEAR COUETTE FLOW PROBLEM

IBAND= 26

NEQ = 82

NO. OF NODES= 35

NO. OF ELEMENTS= 12

NO. OF CORNER NODES= 12

NNVELS= 14

NNCXY= 21

NNPS= 12

SUMMARY OF NODAL COORDINATES

| I | X(I) | Y(I) |
|----|--------|-------|
| 1 | 0.0 | 1.000 |
| 3 | 0.0 | 0.500 |
| 5 | 0.0 | 0.0 |
| 11 | 0.3333 | 1.000 |
| 13 | 0.3333 | 0.500 |
| 15 | 0.3333 | 0.0 |
| 21 | 0.667 | 1.000 |
| 23 | 0.667 | 0.500 |
| 25 | 0.667 | 0.0 |
| 31 | 1.000 | 1.000 |
| 33 | 1.000 | 0.500 |
| 35 | 1.000 | 0.0 |

LISTING OF SYSTEM TOPOLOGY

ELEMENT NUMBER

NODE NUMBERS

| | | | | | | |
|----|----|----|----|----|----|----|
| 1 | 1 | 7 | 13 | 12 | 11 | 6 |
| 2 | 1 | 2 | 3 | 8 | 13 | 7 |
| 3 | 3 | 4 | 5 | 9 | 13 | 8 |
| 4 | | 10 | 15 | 14 | 13 | 9 |
| 5 | 11 | 17 | 23 | 22 | 21 | 16 |
| 6 | 11 | 12 | 13 | 18 | 23 | 17 |
| 7 | 13 | 14 | 15 | 19 | 23 | 18 |
| 8 | 15 | 20 | 25 | 24 | 23 | 19 |
| 9 | 21 | 27 | 33 | 32 | 31 | 26 |
| 10 | 21 | 22 | 23 | 28 | 33 | 27 |
| 11 | 23 | 24 | 25 | 29 | 33 | 28 |
| 12 | 25 | 30 | 35 | 34 | 33 | 29 |

NODES WHERE VELOCITIES ARE SPECIFIED

| I | NODE | U VELOCITY | V VELOCITY |
|----|------|------------|------------|
| 1 | 1 | 1.000 | 0.0 |
| 2 | 5 | 0.0 | 0.0 |
| 3 | 6 | 1.000 | 0.0 |
| 4 | 10 | 0.0 | 0.0 |
| 5 | 11 | 1.000 | 0.0 |
| 6 | 15 | 0.0 | 0.0 |
| 7 | 16 | 1.000 | 0.0 |
| 8 | 20 | 0.0 | 0.0 |
| 9 | 21 | 1.000 | 0.0 |
| 10 | 25 | 0.0 | 0.0 |
| 11 | 26 | 1.000 | 0.0 |
| 12 | 30 | 0.0 | 0.0 |
| 13 | 31 | 1.000 | 0.0 |
| 14 | 35 | 0.0 | 0.0 |

NODES WHERE QX AND QY ARE SPECIFIED

| I | NODE | QX | QY |
|----|------|--------|-----|
| 1 | 7 | 0.0 | 0.0 |
| 2 | 8 | 0.0 | 0.0 |
| 3 | 9 | 0.0 | 0.0 |
| 4 | 12 | 0.0 | 0.0 |
| 5 | 13 | 0.0 | 0.0 |
| 6 | 14 | 0.0 | 0.0 |
| 7 | 17 | 0.0 | 0.0 |
| 8 | 18 | 0.0 | 0.0 |
| 9 | 19 | 0.0 | 0.0 |
| 10 | 22 | 0.0 | 0.0 |
| 11 | 23 | 0.0 | 0.0 |
| 12 | 24 | 0.0 | 0.0 |
| 13 | 27 | 0.0 | 0.0 |
| 14 | 28 | 0.0 | 0.0 |
| 15 | 29 | 0.0 | 0.0 |
| 16 | 22 | 1.333 | 0.0 |
| 17 | 3 | 0.667 | 0.0 |
| 18 | 4 | 1.333 | 0.0 |
| 19 | 32 | -0.333 | 0.0 |
| 20 | 33 | -0.167 | 0.0 |
| 21 | 34 | -0.333 | 0.0 |

NODES WHERE PRESSURE IS SPECIFIED

| I | NODE | PRESSURE |
|----|------|----------|
| 1 | 1 | 4.000 |
| 2 | 3 | 4.000 |
| 3 | 5 | 4.000 |
| 4 | 11 | 3.000 |
| 5 | 13 | 3.000 |
| 6 | 15 | 3.000 |
| 7 | 21 | 2.000 |
| 8 | 23 | 2.000 |
| 9 | 25 | 2.000 |
| 10 | 31 | 1.000 |
| 11 | 33 | 1.000 |
| 12 | 35 | 1.000 |

NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;
 THE V-VELOCITY AT NODES 36 - 70;
 AND THE PRESSURES AT NODES 71 - 82.

THE FIRST SEQUENCE OF 82 NODAL VARIABLES
 REPRESENTS A LINEAR, STEADY STATE SYSTEM;
 WHILE THE SECOND SET OF THE 82 VALUES CORRESPONDS
 TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.

A PRESSURE GRADIENT IN THE HORIZONTAL SHEAR
 DIRECTION OF -3 UNITS IN MAGNITUDE HAS BEEN ADDED
 TO PRODUCE A CURVED VELOCITY PROFILE.

| NCDE NC. | NODE VARIABLES |
|----------|-------------------|
| 1 | 0.1000000000 01 |
| 2 | 0.1031249996D 01 |
| 3 | 0.8750000067D 00 |
| 4 | 0.5312499961D 00 |
| 5 | 0.0 |
| 6 | 0.1000000000 01 |
| 7 | 0.1031249999D 01 |
| 8 | 0.8750000022D 00 |
| 9 | 0.5312499991D 00 |
| 10 | 0.0 |
| 11 | 0.1000000000 01 |
| 12 | 0.1031249999D 01 |
| 13 | 0.8749999998D 00 |
| 14 | 0.5312499995D 00 |
| 15 | 0.0 |
| 16 | 0.1000000000 01 |
| 17 | 0.1031250000 01 |
| 18 | 0.8749999998D 00 |
| 19 | 0.5312499995D 00 |
| 20 | 0.0 |
| 21 | 0.1000000000 01 |
| 22 | 0.1031250002D 01 |
| 23 | 0.8750000014D 00 |
| 24 | 0.5312500020D 00 |
| 25 | 0.0 |
| 26 | 0.1000000000 01 |
| 27 | 0.1031250006D 01 |
| 28 | 0.8749999986D 00 |
| 29 | 0.5312500056D 00 |
| 30 | 0.0 |
| 31 | 0.1000000000 01 |
| 32 | 0.1031250014D 01 |
| 33 | 0.8749999887D 00 |
| 34 | 0.5312500139D 00 |
| 35 | 0.0 |
| 36 | 0.0 |
| 37 | 0.6099426964D-19 |
| 38 | -0.7125187818D-42 |
| 39 | -0.6099426964D-19 |
| 40 | 0.0 |
| 41 | 0.0 |
| 42 | 0.8910283338D-19 |

| | |
|----|-------------------|
| 43 | -0.81332488890-42 |
| 44 | -0.88102833380-19 |
| 45 | 0.0 |
| 46 | 0.0 |
| 47 | 0.19352502520-18 |
| 48 | -0.11149020410-41 |
| 49 | -0.19352502520-18 |
| 50 | 0.0 |
| 51 | 0.0 |
| 52 | 0.47096946920-18 |
| 53 | -0.15718322820-41 |
| 54 | -0.47096946920-18 |
| 55 | 0.0 |
| 56 | 0.0 |
| 57 | 0.11670534580-17 |
| 58 | -0.21868403170-41 |
| 59 | -0.11670534580-17 |
| 60 | 0.0 |
| 61 | 0.0 |
| 62 | 0.29005182730-17 |
| 63 | -0.26715796970-41 |
| 64 | -0.29005182730-17 |
| 65 | 0.0 |
| 66 | 0.0 |
| 67 | 0.20080511240-17 |
| 68 | -0.33565885460-41 |
| 69 | -0.20080511240-17 |
| 70 | 0.0 |
| 71 | 0.40000000000 01 |
| 72 | 0.40000000000 01 |
| 73 | 0.40000000000 01 |
| 74 | 0.30000000000 01 |
| 75 | 0.30000000000 01 |
| 76 | 0.30000000000 01 |
| 77 | 0.20000000000 01 |
| 78 | 0.20000000000 01 |
| 79 | 0.20000000000 01 |
| 80 | 0.10000000000 01 |
| 81 | 0.10000000000 01 |
| 82 | 0.10000000000 01 |

NODE NO.

NODE VARIABLES

1

0.10000000000 01

| | |
|----|-------------------|
| 2 | 0.10288814530 01 |
| 3 | 0.8689584459D 00 |
| 4 | 0.5288644394D 00 |
| 5 | 0.0 |
| 6 | 0.1000000000D 01 |
| 7 | 0.1029118812D 01 |
| 8 | 0.8708250523D 00 |
| 9 | 0.5291035809D 00 |
| 10 | 0.0 |
| 11 | 0.1000000000D 01 |
| 12 | 0.1029300447D 01 |
| 13 | 0.8707893553D 00 |
| 14 | 0.5292824849D 00 |
| 15 | 0.0 |
| 16 | 0.1000000000D 01 |
| 17 | 0.1029562570D 01 |
| 18 | 0.8718195572D 00 |
| 19 | 0.5295482546D 00 |
| 20 | 0.0 |
| 21 | 0.1000000000D 01 |
| 22 | 0.1029726484D 01 |
| 23 | 0.8714843669D 00 |
| 24 | 0.5297056677D 00 |
| 25 | 0.0 |
| 26 | 0.1000000000D 01 |
| 27 | 0.1030036890D 01 |
| 28 | 0.8728957661D 00 |
| 29 | 0.5300233390D 00 |
| 30 | 0.0 |
| 31 | 0.1000000000D 01 |
| 32 | 0.1030173492D 01 |
| 33 | 0.8734199036D 00 |
| 34 | 0.5301609830D 00 |
| 35 | 0.0 |
| 36 | 0.0 |
| 37 | 0.9409332543D-19 |
| 38 | 0.2406344091D-19 |
| 39 | -0.5690421895D-19 |
| 40 | 0.0 |
| 41 | 0.0 |
| 42 | 0.1282493313D-18 |
| 43 | 0.2611072288D-19 |
| 44 | -0.8653878326D-19 |
| 45 | 0.0 |

| | |
|----|-------------------|
| 46 | 0.0 |
| 47 | 0.25383549350-18 |
| 48 | 0.32006190050-19 |
| 49 | -0.19986923340-18 |
| 50 | 0.0 |
| 51 | 0.0 |
| 52 | 0.56134699840-18 |
| 53 | 0.36351135030-19 |
| 54 | -0.49119243710-18 |
| 55 | 0.0 |
| 56 | 0.0 |
| 57 | 0.12819175680-17 |
| 58 | 0.37806257670-19 |
| 59 | -0.12007464690-17 |
| 60 | 0.0 |
| 61 | 0.0 |
| 62 | 0.29622390920-17 |
| 63 | 0.32117633760-19 |
| 64 | -0.29108457190-17 |
| 65 | 0.0 |
| 66 | 0.0 |
| 67 | 0.20082897640-17 |
| 68 | 0.15102713170-19 |
| 69 | -0.19908894340-17 |
| 70 | 0.0 |
| 71 | 0.40000000000 01 |
| 72 | 0.40000000000 01 |
| 73 | 0.40000000000 01 |
| 74 | 0.30000000000 01 |
| 75 | 0.30000000000 01 |
| 76 | 0.30000000000 01 |
| 77 | 0.20000000000 01 |
| 78 | 0.20000000000 01 |
| 79 | 0.20000000000 01 |
| 80 | 0.10000000000 01 |
| 81 | 0.10000000000 01 |
| 82 | 0.10000000000 01 |

STEADY STATE FLUID MECHANICS PROBLEM
(HEAT TRANSFER)
THIS IS A 2-D NONLINEAR PROBLEM

IBAND= 26

NEC=117

NO. OF NODES= 35

NO. OF ELEMENTS= 12

NC. OF CORNER NODES= 12

NNVELS= 20

NNCXY= 15

NNPS= 6

NNTS= 10

NNCZC= 6

NNCZ= 25

SUMMARY OF NODAL COORDINATES

| I | X(I) | Y(I) |
|----|-------|-------|
| 1 | 0.0 | 8.500 |
| 3 | 0.0 | 4.250 |
| 5 | 0.0 | 0.0 |
| 11 | 0.625 | 8.500 |
| 13 | 0.625 | 4.250 |
| 15 | 0.625 | 0.0 |
| 21 | 1.250 | 8.500 |
| 23 | 1.250 | 4.250 |
| 25 | 1.250 | 0.0 |
| 31 | 1.875 | 8.500 |
| 33 | 1.875 | 4.250 |
| 35 | 1.875 | 0.0 |

LISTING OF SYSTEM TOPOLOGY

ELEMENT NUMBER

NODE NUMBERS

| | | | | | | |
|----|----|----|----|----|----|----|
| 1 | 1 | 7 | 13 | 12 | 11 | 6 |
| 2 | 1 | 2 | 3 | 8 | 11 | 7 |
| 3 | 3 | 4 | 5 | 9 | 13 | 8 |
| 4 | 5 | 10 | 15 | 14 | 13 | 9 |
| 5 | 11 | 17 | 23 | 22 | 21 | 16 |
| 6 | 11 | 12 | 13 | 18 | 23 | 17 |
| 7 | 13 | 14 | 15 | 19 | 23 | 18 |
| 8 | 15 | 20 | 25 | 24 | 23 | 19 |
| 9 | 15 | 20 | 25 | 24 | 23 | 19 |
| 10 | 21 | 27 | 33 | 32 | 31 | 26 |
| 11 | 21 | 22 | 23 | 28 | 33 | 27 |
| 12 | 23 | 24 | 25 | 29 | 33 | 28 |
| | 25 | 30 | 35 | 34 | 33 | 29 |

NODES WHERE VELOCITIES ARE SPECIFIED

| I | NODE | U VELOCITY | V VELOCITY |
|---|------|------------|------------|
|---|------|------------|------------|

| | | | |
|----|----|-----|-----|
| 1 | 1 | 0.0 | 0.0 |
| 2 | 2 | 0.0 | 0.0 |
| 3 | 3 | 0.0 | 0.0 |
| 4 | 4 | 0.0 | 0.0 |
| 5 | 5 | 0.0 | 0.0 |
| 6 | 6 | 0.0 | 0.0 |
| 7 | 10 | 0.0 | 0.0 |
| 8 | 11 | 0.0 | 0.0 |
| 9 | 15 | 0.0 | 0.0 |
| 10 | 16 | 0.0 | 0.0 |
| 11 | 20 | 0.0 | 0.0 |
| 12 | 21 | 0.0 | 0.0 |
| 13 | 26 | 0.0 | 0.0 |
| 14 | 27 | 0.0 | 0.0 |
| 15 | 28 | 0.0 | 0.0 |
| 16 | 33 | 0.0 | 0.0 |
| 17 | 34 | 0.0 | 0.0 |
| 18 | 35 | 0.0 | 0.0 |
| 19 | 35 | 0.0 | 0.0 |
| 20 | 35 | 0.0 | 0.0 |

NODES WHERE QX AND QY ARE SPECIFIED

| I | NODE | QX | QY |
|----|------|-----|-----|
| 1 | 7 | 0.0 | 0.0 |
| 2 | 8 | 0.0 | 0.0 |
| 3 | 9 | 0.0 | 0.0 |
| 4 | 12 | 0.0 | 0.0 |
| 5 | 13 | 0.0 | 0.0 |
| 6 | 14 | 0.0 | 0.0 |
| 7 | 17 | 0.0 | 0.0 |
| 8 | 18 | 0.0 | 0.0 |
| 9 | 19 | 0.0 | 0.0 |
| 10 | 22 | 0.0 | 0.0 |
| 11 | 23 | 0.0 | 0.0 |
| 12 | 24 | 0.0 | 0.0 |
| 13 | 27 | 0.0 | 0.0 |
| 14 | 28 | 0.0 | 0.0 |
| 15 | 29 | 0.0 | 0.0 |

NODES WHERE PRESSURE IS SPECIFIED

| I | NODE | PRESSURE |
|---|------|-------------|
| 1 | 1 | 1014000.000 |
| 2 | 2 | 1014000.000 |
| 3 | 3 | 1014000.000 |
| 4 | 5 | 1014000.000 |
| 5 | 31 | 1014000.000 |
| 6 | 33 | 1014000.000 |
| 7 | 35 | 1014000.000 |

NODES WHERE TEMPERATURE IS SPECIFIED

| I | NODE | TEMPERATURE |
|----|------|-------------|
| 1 | 1 | 25.000 |
| 2 | 2 | 25.000 |
| 3 | 3 | 25.000 |
| 4 | 4 | 25.000 |
| 5 | 5 | 25.000 |
| 6 | 12 | 20.000 |
| 7 | 22 | 20.000 |
| 8 | 23 | 20.000 |
| 9 | 24 | 20.000 |
| 10 | 35 | 20.000 |

NODES WHERE QZC IS SPECIFIED

| I | NODE | QZC |
|---|------|-----|
| 1 | 4 | 0.0 |
| 2 | 5 | 0.0 |
| 3 | 6 | 0.0 |
| 4 | 7 | 0.0 |
| 5 | 8 | 0.0 |
| 6 | 9 | 0.0 |

NODES WHERE HEAT FLUX QZ IS SPECIFIED

| I | NODE | HEAT FLUX |
|----|------|-----------|
| 1 | 6 | 0.0 |
| 2 | 7 | 0.00 |
| 3 | 8 | 0.00 |
| 4 | 9 | 0.00 |
| 5 | 10 | 0.00 |
| 6 | 11 | 0.00 |
| 7 | 12 | 0.00 |
| 8 | 13 | 0.00 |
| 9 | 14 | 0.00 |
| 10 | 15 | 0.00 |
| 11 | 16 | 0.00 |
| 12 | 17 | 0.00 |
| 13 | 18 | 0.00 |
| 14 | 19 | 0.00 |
| 15 | 20 | 0.00 |
| 16 | 21 | 0.00 |
| 17 | 22 | 0.00 |
| 18 | 23 | 0.00 |
| 19 | 24 | 0.00 |
| 20 | 25 | 0.00 |
| 21 | 26 | 0.00 |
| 22 | 27 | 0.00 |
| 23 | 28 | 0.00 |
| 24 | 29 | 0.00 |
| 25 | 30 | 0.0 |

NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;
 THE V-VELOCITY AT NODES 36 - 70;
 THE PRESSURE AT NODES 71 - 82;
 AND THE TEMPERATURE AT NODES 83 - 117.

THE SPECIFIED WALL PRESSURES
 ARE NORMALIZED TO ONE (1) ATMOSPHERE,
 THAT IS, 1014000 DYNES/SQ.CM
 (ALL PARAMETER VALUES ARE IN CGS UNITS).

THE FIRST SEQUENCE OF 117 NODAL VARIABLES
 REPRESENTS A LINEAR, STEADY STATE SYSTEM;
 WHILE THE SECOND SET OF THE 117 VALUES CORRESPONDS
 TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.

| NODE NO. | NODE VARIABLE |
|----------|---------------|
| 1 | 0.0 |
| 2 | 0.0 |
| 3 | 0.0 |
| 4 | 0.0 |

| | |
|----|-------------------|
| 5 | 0.0 |
| 6 | 0.0 |
| 7 | -0.1594580172D 01 |
| 8 | -0.7882672787D 01 |
| 9 | -0.1594580172D 01 |
| 10 | 0.0 |
| 11 | 0.0 |
| 12 | -0.3649507488D 00 |
| 13 | -0.1203896378D 02 |
| 14 | -0.3649507488D 00 |
| 15 | 0.0 |
| 16 | 0.0 |
| 17 | -0.1216689153D 00 |
| 18 | -0.9223308785D 01 |
| 19 | -0.1216689153D 00 |
| 20 | 0.0 |
| 21 | 0.0 |
| 22 | -0.1605186515D 01 |
| 23 | -0.1070997583D 02 |
| 24 | -0.1605186515D 01 |
| 25 | 0.0 |
| 26 | 0.0 |
| 27 | -0.1686613562D 01 |
| 28 | -0.6458410240D 01 |
| 29 | -0.1686613562D 01 |
| 30 | 0.0 |
| 31 | 0.0 |
| 32 | 0.0 |
| 33 | 0.0 |
| 34 | 0.0 |
| 35 | 0.0 |
| 36 | 0.0 |
| 37 | 0.0 |
| 38 | 0.0 |
| 39 | 0.0 |
| 40 | 0.0 |
| 41 | 0.0 |
| 42 | 0.3287813820D 00 |
| 43 | -0.8671505203D-23 |
| 44 | -0.3287813820D 00 |
| 45 | 0.0 |
| 46 | 0.0 |
| 47 | 0.4841816743D 00 |
| 48 | -0.1212888826D-22 |

| | |
|----|-------------------|
| 49 | -0.48418181430 00 |
| 50 | 0.0 |
| 51 | 0.0 |
| 52 | 0.28532060500 00 |
| 53 | 0.42109196680-23 |
| 54 | -0.28532060500 00 |
| 55 | 0.0 |
| 56 | 0.0 |
| 57 | 0.15892437070-01 |
| 58 | 0.92415574840-23 |
| 59 | -0.15892437070-01 |
| 60 | 0.0 |
| 61 | 0.0 |
| 62 | -0.43460776970-01 |
| 63 | 0.11208359430-22 |
| 64 | 0.43460776970-01 |
| 65 | 0.0 |
| 66 | 0.0 |
| 67 | 0.0 |
| 68 | 0.0 |
| 69 | 0.0 |
| 70 | 0.0 |
| 71 | 0.10140000000 07 |
| 72 | 0.10140000000 07 |
| 73 | 0.10140000000 07 |
| 74 | 0.10140370640 07 |
| 75 | 0.10142256570 07 |
| 76 | 0.10140370640 07 |
| 77 | 0.10139803130 07 |
| 78 | 0.10138750670 07 |
| 79 | 0.10139803130 07 |
| 80 | 0.10140000000 07 |
| 81 | 0.10140000000 07 |
| 82 | 0.10140000000 07 |
| 83 | 0.25000000000 02 |
| 84 | 0.25000000000 02 |
| 85 | 0.25000000000 02 |
| 86 | 0.25000000000 02 |
| 87 | 0.25000000000 02 |
| 88 | 0.24166666670 02 |
| 89 | 0.24166666670 02 |
| 90 | 0.24166666670 02 |
| 91 | 0.24166666670 02 |
| 92 | 0.24166666670 02 |

| | |
|-----|-----------------|
| 93 | 0.233333333D 02 |
| 94 | 0.233333333D 02 |
| 95 | 0.233333333D 02 |
| 96 | 0.233333333D 02 |
| 97 | 0.233333333D 02 |
| 98 | 0.225000000D 02 |
| 99 | 0.225000000D 02 |
| 100 | 0.225000000D 02 |
| 101 | 0.225000000D 02 |
| 102 | 0.225000000D 02 |
| 103 | 0.216666666D 02 |
| 104 | 0.216666666D 02 |
| 105 | 0.216666666D 02 |
| 106 | 0.216666666D 02 |
| 107 | 0.216666666D 02 |
| 108 | 0.208333333D 02 |
| 109 | 0.208333333D 02 |
| 110 | 0.208333333D 02 |
| 111 | 0.208333333D 02 |
| 112 | 0.208333333D 02 |
| 113 | 0.200000000D 02 |
| 114 | 0.200000000D 02 |
| 115 | 0.200000000D 02 |
| 116 | 0.200000000D 02 |
| 117 | 0.200000000D 02 |

| NODE NO. | NODE VARIABLE |
|----------|-------------------|
| 1 | 0.0 |
| 2 | 0.0 |
| 3 | 0.0 |
| 4 | 0.0 |
| 5 | 0.0 |
| 6 | 0.0 |
| 7 | -0.1671967327D 01 |
| 8 | -0.8146388287D 01 |
| 9 | -0.1667709024D 01 |
| 10 | 0.0 |
| 11 | 0.0 |
| 12 | -0.3529066201D 00 |
| 13 | -0.1267086943D 02 |
| 14 | -0.3464796968D 00 |
| 15 | 0.0 |
| 16 | 0.0 |

| | |
|----|-------------------|
| 17 | -0.95541551550-01 |
| 18 | -0.96623861040 01 |
| 19 | -0.88746546530-01 |
| 20 | 0.0 |
| 21 | 0.0 |
| 22 | -0.16437945500 01 |
| 23 | -0.11188367610 02 |
| 24 | -0.16393742270 01 |
| 25 | 0.0 |
| 26 | 0.0 |
| 27 | -0.16663660230 01 |
| 28 | -0.68680556020 01 |
| 29 | -0.16597517820 01 |
| 30 | 0.0 |
| 31 | 0.0 |
| 32 | 0.0 |
| 33 | 0.0 |
| 34 | 0.0 |
| 35 | 0.0 |
| 36 | 0.0 |
| 37 | 0.0 |
| 38 | 0.0 |
| 39 | 0.0 |
| 40 | 0.0 |
| 41 | 0.0 |
| 42 | 0.39280003440 00 |
| 43 | 0.28717151270-03 |
| 44 | -0.39167605910 00 |
| 45 | 0.0 |
| 46 | 0.0 |
| 47 | 0.59303609440 00 |
| 48 | 0.61262434900-03 |
| 49 | -0.59071993840 00 |
| 50 | 0.0 |
| 51 | 0.0 |
| 52 | 0.37670797430 00 |
| 53 | 0.99603754060-03 |
| 54 | -0.37314213780 00 |
| 55 | 0.0 |
| 56 | 0.0 |
| 57 | 0.63827156120-01 |
| 58 | 0.13760499030-02 |
| 59 | -0.59875228560-01 |
| 60 | 0.0 |

| | |
|-----|-------------------|
| 61 | 0.0 |
| 62 | -0.2429545074D-01 |
| 63 | 0.1210812751D-02 |
| 64 | 0.2677263620D-01 |
| 65 | 0.0 |
| 66 | 0.0 |
| 67 | 0.0 |
| 68 | 0.0 |
| 69 | 0.0 |
| 70 | 0.0 |
| 71 | 0.1014000000D 07 |
| 72 | 0.1014000000D 07 |
| 73 | 0.1014000000D 07 |
| 74 | 0.1014038420D 07 |
| 75 | 0.1014261630D 07 |
| 76 | 0.1014038452D 07 |
| 77 | 0.1013999776D 07 |
| 78 | 0.1013884020D 07 |
| 79 | 0.1014000754D 07 |
| 80 | 0.1014000000D 07 |
| 81 | 0.1014000000D 07 |
| 82 | 0.1014000000D 07 |
| 83 | 0.2500000000D 02 |
| 84 | 0.2500000000D 02 |
| 85 | 0.2500000000D 02 |
| 86 | 0.2500000000D 02 |
| 87 | 0.2500000000D 02 |
| 88 | 0.2411490471D 02 |
| 89 | 0.2427187487D 02 |
| 90 | 0.2463978352D 02 |
| 91 | 0.2423258019D 02 |
| 92 | 0.2364452284D 02 |
| 93 | 0.2323056484D 02 |
| 94 | 0.2350586481D 02 |
| 95 | 0.2423194982D 02 |
| 96 | 0.2343829368D 02 |
| 97 | 0.2227636650D 02 |
| 98 | 0.2241562105D 02 |
| 99 | 0.2270150631D 02 |
| 100 | 0.2339548232D 02 |
| 101 | 0.2262750919D 02 |
| 102 | 0.2151348362D 02 |
| 103 | 0.2159431099D 02 |
| 104 | 0.2185937901D 02 |

| | |
|-----|------------------|
| 105 | 0.22485905170 02 |
| 106 | 0.21795465440 02 |
| 107 | 0.20708330220 02 |
| 108 | 0.20806085770 02 |
| 109 | 0.20955083380 02 |
| 110 | 0.21294435330 02 |
| 111 | 0.20918293520 02 |
| 112 | 0.20370358320 02 |
| 113 | 0.20000000000 02 |
| 114 | 0.20000000000 02 |
| 115 | 0.20000000000 02 |
| 116 | 0.20000000000 02 |
| 117 | 0.20000000000 02 |

THE KINEMATIC VISCOSITY OF FLUID 50-HB-3520 AT 20 DEGREES C. =
10.90 SC.CM/SEC

THE DENSITY OF 50-HB-3520 AT 20 DEGREES C. = 1.0596 GM/CC

THE COEFF. OF THERMAL EXPANSION OF 50-HB-3520 AT 20 DEGREES C. =
0.002278/DEGREE C.

THE THERMAL DIFFUSIVITY OF 50-HB-3520 AT 20 DEGREES C. =
0.00103 SQ.CM/SEC

THE GRASHOF NUMBER (GR(L)) = $(G \cdot B \cdot L^3 \cdot (T_H - T_C)) / V^2$ = 946.4

THE U VELOCITY FORCING FUNCTION, $G \cdot B \cdot T(\text{INITIAL})$, = 65.962 CM/SQ.S

STEADY STATE COUETTE FLOW PROBLEM

```

C      IMPLICIT REAL*8(A-H,O-Z,$)
C      THE U VELOCITY IS IN THE FIRST NN POSITIONS OF T(I) (NN=NO. OF NODES
C      THE V VELOCITY IS IN THE SECOND NN POSITIONS OF T I.E. T(I+NN)
C      THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF T I.E. T(I+NN+NN)
C      THERE ARE NNCN PRESSURE NODES
C      DATA NREAD/5/
C      DATA NWRITE/6/
C      DATA STOP/'STOP' /
C      DIMENSION TM(82,82), YC(82), NODE(82,6), NVS(82), NCP(82), NCN(82)
C      DIMENSION T(82), NVIS(82), NPS(82), Q(82)
C      DIMENSION NQS(82), T1(82)
C      DIMENSION TM$(15), I5), N(15), NQIS(82)
C      DIMENSION RHS(82)
C      DIMENSION CN RP$(6), ZP$(6)
C      DIMENSION XC$(3), YC$(3), WKAREA(12000)
C      THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE
C      LINE 014 TO 0151 ARE INPUT AND VERIFICATION OF ALL DATA
C      THIS WOULD BE PART OF ANY FINITE ELEMENT PROGRAM
C      READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CORNER NODES
C      READ (NREAD,3800) NN,NE,NNCN
C      WRITE(NWRITE,1094)
C      INITIALIZE ALL PARAMETERS
C      MM = 2*NN+NNCN
C      DO 200 I=1,MM
C      XC(I) = 0.DO
C      COUT0001
C      COUT0002
C      COUT0003
C      COUT0004
C      COUT0005
C      COUT0006
C      COUT0007
C      COUT0008
C      COUT0009
C      COUT0010
C      COUT0011
C      COUT0012
C      COUT0013
C      COUT0014
C      COUT0015
C      COUT0016
C      COUT0017
C      COUT0018
C      COUT0019
C      COUT0020
C      COUT0021
C      COUT0022
C      COUT0023
C      COUT0024
C      COUT0025
C      COUT0026
C      COUT0027
C      COUT0028
C      COUT0029
C      COUT0030
C      COUT0031
C      COUT0032
C      COUT0033
C      COUT0034
C      COUT0035
C      COUT0036

```

```

YC(I) = 0.00
NVS(I) = 0
NCN(I) = 0
NCP(I) = 0
NPS(I) = 0
NQS(I) = 0
C
DO 200 J=1,6
NODE(I,J) = 0
CONTINUE
200
C
DO 300 I=1,MM
T1(I) = 0.00
T(I) = 0.00
NVIS(I) = 0
Q(I) = 0.00
NQIS(I) = 0
RHS(I) = 0.00
C
DO 300 J=1,MM
TM(I,J) = 0.00
CONTINUE
300
C
DO 400 I=1,15
N(I) = 0
C
DO 400 J=1,15
TM$(I,J) = 0.00
CONTINUE
400
C
DO 500 I=1,6
RP$(I) = 0.00
ZP$(I) = 0.00
CONTINUE
500
C
DO 600 I=1,3
XC$(I) = 0.00
YC$(I) = 0.00
CONTINUE
600
C
READ NODE NUMBERS AND COORDINATES
C
C
C
DO 700 J=1,NN

```

```

COUT00037
COUT00038
COUT00039
COUT00040
COUT00041
COUT00042
COUT00043
COUT00044
COUT00045
COUT00046
COUT00047
COUT00048
COUT00049
COUT00050
COUT00051
COUT00052
COUT00053
COUT00054
COUT00055
COUT00056
COUT00057
COUT00058
COUT00059
COUT00060
COUT00061
COUT00062
COUT00063
COUT00064
COUT00065
COUT00066
COUT00067
COUT00068
COUT00069
COUT00070
COUT00071
COUT00072
COUT00073
COUT00074
COUT00075
COUT00076
COUT00077
COUT00078
COUT00079
COUT00080
COUT00081
COUT00082
COUT00083
COUT00084

```

```

C      READ (NREAD,3900) WORD,I,XC(I),YC(I)
C      IF (WORD.EQ.STOP) GO TO 800
C      NCN(J) = I
C      700 CONTINUE
C
C      800 NNCN = J-1
C
C      THE ARRAY NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2.)
C      THUS PRESSURE NODES ARE LABELED AS CORNER NODES ARE INPUTTED
C      WHEN ONE INPUTS A GLOBAL CORNER NODE FOR J
C
C      DO 900 J=1,NNCN
C      NCP(NCN(J)) = J+NN+NN
C      900 CONTINUE
C
C      SYSTEM TOPOLOGY( ELEMENT NO. AND NODE NUMBERS IN
C      COUNTER-CLOCKWISE FASHION STARTING AT ANY CORNER NODE
C      ALWAYS COUNT FROM UPPER LEFT HAND CORNER
C
C      DO 1000 I=1,NE
C      READ (NREAD,4000) J,NODE(J,1),NODE(J,2),NODE(J,3),NODE(J,4),NODE(J,5),NODE(J,6)
C      1000 CONTINUE
C
C      MAXDIF = 0
C
C      DO 1100 I=1,NE
C
C      DO 1100 J=1,6
C
C      DO 1100 K=1,6
C      LL = IABS(NODE(I,J)-NODE(I,K))
C      IF (LL.GT.MAXDIF) MAXDIF=LL
C      IBAND = 2*(MAXDIF+1)
C      NEQ = 2*NN+NNCN
C      1100 CONTINUE
C
C      WRITE (NWRITE,3700) IBAND,NEQ
C
C      READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED
C
C      DO 1200 I=1,MM
C      READ (NREAD,3900) WORD,NVELS,VELU,VELV
C      IF (WORD.EQ.STOP) GO TO 1300
C
C      COUT0085
C      COUT0086
C      COUT0087
C      COUT0088
C      COUT0089
C      COUT0090
C      COUT0091
C      COUT0092
C      COUT0093
C      COUT0094
C      COUT0095
C      COUT0096
C      COUT0097
C      COUT0098
C      COUT0099
C      COUT0100
C      COUT0101
C      COUT0102
C      COUT0103
C      COUT0104
C      COUT0105
C      COUT0106
C      COUT0107
C      COUT0108
C      COUT0109
C      COUT0110
C      COUT0111
C      COUT0112
C      COUT0113
C      COUT0114
C      COUT0115
C      COUT0116
C      COUT0117
C      COUT0118
C      COUT0119
C      COUT0120
C      COUT0121
C      COUT0122
C      COUT0123
C      COUT0124
C      COUT0125
C      COUT0126
C      COUT0127
C      COUT0128
C      COUT0129
C      COUT0130
C      COUT0131
C      COUT0132

```

```

      NVS(I) = NVELS
      T(NVS(I)) = VELU
      T(NVS(I)+NN) = VELV
1200 CONTINUE
C
      COUNT NODES HAVING SPECIFIED VELOCITIES
C
1300 NVELS = I-1
C
      READ QX AND QY VALUES AT INTERNAL NODES
C
      DO 1400 I=1,NN
      READ (NREAD,3900) WORD,NQXY,QXNS,QYNS
      IF (WORD.EQ.STOP) GO TO 1500
      NQS(I) = NQXY
      Q(NQS(I)) = QXNS
      Q(NQS(I)+NN) = QYNS
1400 CCNTINUE
C
      COUNT NODES HAVING SPECIFIED QX AND QY
C
1500 NNQXY = I-1
C
      READ NODE NUMBER AND PRESSURE WHERE SPECIFIED
C
      DO 1600 I=1,NN
      READ (NREAD,4100) WORD,NP,PNP
      IF (WORD.EQ.STOP) GO TO 1700
      NPS(I) = NP
      T(NCP(NPS(I))) = PNP
1600 CONTINUE
C
      COUNT BOUNDARY NODES WHERE PRESSURE SPECIFIED
C
1700 NNPS = I-1
C
      NQIS IS A LIST OF THE INDICES OF KNOWN QX,QY
C
      DO 2000 I=1,NNQXY
      NQIS(I) = NQS(I)
      NQIS(I+NNQXY) = NQS(I)+NN
2000 CCNTINUE
C

```

```

COUT01133
COUT01134
COUT01135
COUT01136
COUT01137
COUT01138
COUT01139
COUT01140
COUT01141
COUT01142
COUT01143
COUT01144
COUT01145
COUT01146
COUT01147
COUT01148
COUT01149
COUT01150
COUT01151
COUT01152
COUT01153
COUT01154
COUT01155
COUT01156
COUT01157
COUT01158
COUT01159
COUT01160
COUT01161
COUT01162
COUT01163
COUT01164
COUT01165
COUT01166
COUT01167
COUT01168
COUT01169
COUT01170
COUT01171
COUT01172
COUT01173
COUT01174
COUT01175
COUT01176
COUT01177
COUT01178
COUT01179
COUT01180

```

```

C      NVIS IS A LIST OF KNOWN VELOCITY AND PRESSURE INDICES
C
C      DO 2200 I=1,NNVELS
C      NVIS(I) = NVS(I)
C      NVIS(I+NNVELS) = NVS(I)+NN
C      CONTINUE
C
C      DO 2300 J=1,NNPS
C      NVIS(2*NNVELS+J) = NCP(NPS(J))
C      CONTINUE
C
C      NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED
C      NNHC = 0
C
C      NTOTQ=TOTAL NUMBER OF KNOWN QX,QY
C      NTOTQ = 2*NNQXY
C
C      NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, AND PRESSURES
C      NTOTVP = 2*NNVELS+NNPS
C
C      PRINT ALL INPUT DATA
C
C      WRITE (NWRITE,4300) NN,NE,NNCN
C      WRITE (NWRITE,4400) NNVELS
C      WRITE (NWRITE,4500) NNQXY
C      WRITE (NWRITE,4600) NNPS
C      WRITE (NWRITE,4700)
C
C      DO 2400 I=1,NNCN
C      WRITE (NWRITE,4800) NCN(I),XC(NCN(I)),YC(NCN(I))
C      CONTINUE
C
C      WRITE (NWRITE,4900)
C
C      DO 2500 I=1,NE
C      WRITE (NWRITE,5000) I,NODE(I,1),NODE(I,2),NODE(I,3),NODE(I,4),NODE
C      1(I,5),NODE(I,6)
C      CONTINUE
C
C      WRITE (NWRITE,5100)
C
C      DO 2600 I=1,NNVELS
C      WRITE (NWRITE,5200) I,NVS(I),T(NVS(I)),T(NVS(I)+NN)

```

```

2600 CONTINUE
C
WRITE (NWRITE,5300)
C
DO 2700 I=1,NNQXY
WRITE (NWRITE,5200) I,NQS(I),Q(NQS(I)),Q(NQS(I))+NN)
CONTINUE
2700
C
WRITE (NWRITE,5400)
C
DO 2800 I=1,NNNPS
WRITE (NWRITE,5600) I,NPS(I),T(NCP(NPS(I)))
CONTINUE
2800
C
WRITE(NWRITE,1090)
WRITE(NWRITE,1092)
WRITE(NWRITE,1093)
CONTINUE
2840
C
DO 2850 I=1,MM
RHS(I) = 0.00
DO 2850 J=1,MM
TM(I,J) = 0.00
CONTINUE
2850
C
DO 2860 I=1,15
DO 2860 J=1,15
TM$(I,J) = 0.00
CONTINUE
2860
C
C
C
C
C
C
END OF INPUT AND VERIFICATION ROUTINE

DO 3200 K=1,NE
N1=NODE(K,1)
N2=NODE(K,2)
N3=NODE(K,3)
N4=NODE(K,4)
N5=NODE(K,5)
N6=NODE(K,6)
N7=NODE(K,1)+NN
N8=NODE(K,2)+NN
N9=NODE(K,3)+NN
N10=NODE(K,4)+NN
N11=NODE(K,5)+NN
N12=NODE(K,6)+NN
N13=NCP(NODE(K,1))
N14=NCP(NODE(K,3))
N15=NCP(NODE(K,5))

```

```

XC(NODE(K,1))
XC(NODE(K,3))
XC(NODE(K,5))
XC(NODE(K,1))
XC(NODE(K,3))
XC(NODE(K,5))
DO
  AA = 1.
  AA1 = XC$(1)*XC$(2)
  AA2 = XC$(1)*XC$(3)
  AA3 = XC$(1)*XC$(5)
  BB1 = XC$(2)*XC$(3)
  BB2 = XC$(2)*XC$(5)
  BB3 = XC$(3)*XC$(5)
  CC1 = XC$(1)*XC$(2)
  CC2 = XC$(1)*XC$(3)
  CC3 = XC$(1)*XC$(5)
  DEL = XC$(1)*XC$(2)*XC$(3)*XC$(5)
  1$CONST = (1.D0*AA)
  D1 = -B1/6.D0
  D2 = -B2/6.D0
  D3 = -B3/6.D0
  DE1 = -C1/6.D0
  DE2 = -C2/6.D0
  DE3 = -C3/6.D0
  FF1 = B1/2520.D0
  FF2 = B2/2520.D0
  FF3 = B3/2520.D0
  GG1 = C1/2520.D0
  GG2 = C2/2520.D0
  GG3 = C3/2520.D0
  UU1 = T1(N1)
  UU2 = T1(N2)
  UU3 = T1(N3)
  UU4 = T1(N4)
  UU5 = T1(N5)
  UU6 = T1(N6)
  UU7 = T1(N7)
  UU8 = T1(N8)
  UU9 = T1(N9)
  UU10 = T1(N10)
  UU11 = T1(N11)
  UU12 = T1(N12)
  VV1 = T1(N1)
  VV2 = T1(N2)
  VV3 = T1(N3)
  VV4 = T1(N4)
  VV5 = T1(N5)
  VV6 = T1(N6)
  VV7 = T1(N7)
  VV8 = T1(N8)
  VV9 = T1(N9)
  VV10 = T1(N10)
  VV11 = T1(N11)
  VV12 = T1(N12)
  TT1 = T1(N1)
  TT2 = T1(N2)
  TT3 = T1(N3)
  TT4 = T1(N4)
  TT5 = T1(N5)
  TT6 = T1(N6)
  TT7 = T1(N7)
  TT8 = T1(N8)
  TT9 = T1(N9)
  TT10 = T1(N10)
  TT11 = T1(N11)
  TT12 = T1(N12)
  0.75D0*(B1*B1+C1*C1)*CONST
  (B1*B2+C1*C2)*CONST
  -TM$(1,2)*0.25D0
  0.

```

```

TM$(1,6) = (B1*B3+C1*C3)*CONST
TM$(1,5) = -TM$(1,6)*.25D0
TM$(3,3) = .75D0*(B2*B3+C2*C3)*CONST
TM$(3,4) = (B2*B3+C2*C3)*CONST
TM$(3,5) = -TM$(3,4)*.25D0
TM$(2,1) = TM$(1,2)
TM$(2,3) = TM$(1,2)
TM$(2,2) = 8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)
TM$(2,5) = 2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)
TM$(2,6) = 0.
TM$(3,1) = TM$(1,3)
TM$(3,2) = TM$(2,3)
TM$(3,3) = 0.
TM$(3,6) = TM$(3,6)
TM$(5,5) = .75D0*(B3*B3+C3*C3)*CONST
TM$(4,1) = .TM$(1,4)
TM$(4,2) = TM$(2,4)
TM$(4,3) = TM$(3,4)
TM$(4,4) = 8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)
TM$(4,5) = TM$(3,4)
TM$(4,6) = TM$(1,6)+TM$(3,4)+2.D0*TM$(1,2)+4.D0/3.D0*TM$(5,5)
TM$(5,1) = TM$(2,5)
TM$(5,2) = TM$(3,5)
TM$(5,3) = TM$(4,5)
TM$(5,4) = TM$(4,5)
TM$(5,6) = TM$(1,6)
TM$(6,1) = TM$(2,6)
TM$(6,2) = TM$(2,6)
TM$(6,4) = TM$(4,6)
TM$(6,5) = TM$(5,6)
TM$(6,6) = 8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)

      BEGIN INPUT OF NONLINEAR TERMS
1-TM$(1,1)=TM$(1,1)
2-(78.D0*U1+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1
3*G1
1-TM$(2,1)=TM$(2,1)
2-(48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1
3V6)*G1
1-TM$(3,1)=TM$(3,1)
2-(-9.D0*U1-32.D0*U2-18.D0*U3-16.D0*U4+11.D0*U5-20.D0*U6)*F1
36)*G1
TM$(4,1)=TM$(4,1)

```

```

COUT0325
COUT0326
COUT0327
COUT0328
COUT0329
COUT0330
COUT0331
COUT0332
COUT0333
COUT0334
COUT0335
COUT0336
COUT0337
COUT0338
COUT0339
COUT0340
COUT0341
COUT0342
COUT0343
COUT0344
COUT0345
COUT0346
COUT0347
COUT0348
COUT0349
COUT0350
COUT0351
COUT0352
COUT0353
COUT0354
COUT0355
COUT0356
COUT0357
COUT0358
COUT0359
COUT0360
COUT0361
COUT0362
COUT0363
COUT0364
COUT0365
COUT0366
COUT0367
COUT0368
COUT0369
COUT0370
COUT0371
COUT0372

```

CC

```

1- (12. D0*U1+16. D0*U2-16. D0*U3-96. D0*U4-16. D0*U5+16. D0*U6)*F1      COUT0373
2- (12. D0*V1+16. D0*V2-16. D0*V3-96. D0*V4-16. D0*V5+16. D0*V6)*F1      COUT0374
36)*G1                                     COUT0375
TM$(5,1)=TM$(5,1)                         COUT0376
1- (-9. D0*U1-20. D0*U2+11. D0*U3-16. D0*U4-18. D0*U5-32. D0*U6)*F1      COUT0377
2- (-9. D0*V1-20. D0*V2+11. D0*V3-16. D0*V4-18. D0*V5-32. D0*V6)*F1      COUT0378
36)*G1                                     COUT0379
TM$(6,1)=TM$(6,1)                         COUT0380
1- (48. D0*U1+80. D0*U2-20. D0*U3+16. D0*U4-32. D0*U5+160. D0*U6)*F1      COUT0381
2- (48. D0*V1+80. D0*V2-20. D0*V3+16. D0*V4-32. D0*V5+160. D0*V6)*F1      COUT0382
3V6)*G1                                     COUT0383
TM$(1,2)=TM$(1,2)                         COUT0384
1- (24. D0*U1-32. D0*U2-16. D0*U3-48. D0*U4+4. D0*U5-16. D0*U6)*F1      COUT0385
2- (24. D0*V1-32. D0*V2-16. D0*V3-48. D0*V4+4. D0*V5-16. D0*V6)*F1      COUT0386
3)*G1                                     COUT0387
4D0*U5+48. D0*U6)*F2                      COUT0388
516. D0*V4-16. D0*V5+48. D0*V6)*G2      COUT0389
TM$(2,2)=TM$(2,2)                         COUT0390
1- (-32. D0*U1+384. D0*U2+48. D0*U3+192. D0*U4-48. D0*U5+128. D0*U6)*F1      COUT0391
2- (-32. D0*V1+384. D0*V2+48. D0*V3+192. D0*V4-48. D0*V5+128. D0*V6)*F1      COUT0392
3D0*U6)*G1                                 COUT0393
4D0*U5+192. D0*U6)*F2                      COUT0394
5128. D0*V4-48. D0*V5+192. D0*V6)*G2      COUT0395
TM$(3,2)=TM$(3,2)                         COUT0396
1- (-16. D0*U1+48. D0*U2+120. D0*U3+48. D0*U4-16. D0*U5-16. D0*U6)*F1      COUT0397
2- (-16. D0*V1+48. D0*V2+120. D0*V3+48. D0*V4-16. D0*V5-16. D0*V6)*F1      COUT0398
3V6)*G1                                     COUT0399
40*U5-48. D0*U6)*F2                      COUT0400
516. D0*V4+4. D0*V5-48. D0*V6)*G2      COUT0401
TM$(4,2)=TM$(4,2)                         COUT0402
1- (-48. D0*U1+192. D0*U2+48. D0*U3+384. D0*U4-32. D0*U5+128. D0*U6)*F1      COUT0403
2- (-48. D0*V1+192. D0*V2+48. D0*V3+384. D0*V4-32. D0*V5+128. D0*V6)*F1      COUT0404
3D0*U6)*G1                                 COUT0405
46. D0*U5+128. D0*U6)*F2                      COUT0406
5+128. D0*V4-16. D0*V5+128. D0*V6)*G2      COUT0407
TM$(5,2)=TM$(5,2)                         COUT0408
1- (4. D0*U1-48. D0*U2-16. D0*U3-32. D0*U4+24. D0*U5-16. D0*U6)*F1      COUT0409
2- (4. D0*V1-48. D0*V2-16. D0*V3-32. D0*V4+24. D0*V5-16. D0*V6)*F1      COUT0410
3)*G1                                     COUT0411
40*U5-32. D0*U6)*F2                      COUT0412
56. D0*V4+24. D0*V5-32. D0*V6)*G2      COUT0413
TM$(6,2)=TM$(6,2)                         COUT0414
1- (-16. D0*U1+128. D0*U2-16. D0*U3+128. D0*U4-16. D0*U5+128. D0*U6)*F1      COUT0415
2- (-16. D0*V1+128. D0*V2-16. D0*V3+128. D0*V4-16. D0*V5+128. D0*V6)*F1      COUT0416
3D0*U6)*G1                                 COUT0417
4D0*U5+384. D0*U6)*F2                      COUT0418
5128. D0*V4-32. D0*V5+384. D0*V6)*G2      COUT0419
TM$(1,3)=TM$(1,3)                         COUT0420

```

```

1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2      COUT0421
2-(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*V6)*F2      COUT0422
3)*G2                             COUT0423
TM$(2,3)=TM$(2,3)                COUT0424
1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2      COUT0425
2-(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0*V6)*F2      COUT0426
3)*G2                             COUT0427
TM$(3,3)=TM$(3,3)                COUT0428
1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2      COUT0429
2-(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6)*F2      COUT0430
3)*G2                             COUT0431
TM$(4,3)=TM$(4,3)                COUT0432
1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2      COUT0433
2-(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0*V6)*F2      COUT0434
3)*G2                             COUT0435
TM$(5,3)=TM$(5,3)                COUT0436
1-(-11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2      COUT0437
2-(-11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6)*F2      COUT0438
3)*G2                             COUT0439
TM$(6,3)=TM$(6,3)                COUT0440
1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2      COUT0441
2-(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*V6)*F2      COUT0442
3)*G2                             COUT0443
TM$(1,4)=TM$(1,4)                COUT0444
1-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2      COUT0445
2-(-24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6)*F2      COUT0446
3)*G2                             COUT0447
TM$(1,4)=TM$(1,4)                COUT0448
4)*G2                             COUT0449
58.D0*U5-16.D0*U6)*F3            COUT0450
58.D0*V4+4.D0*V5-16.D0*V6)*G3   COUT0451
TM$(2,4)=TM$(2,4)                COUT0452
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F2      COUT0453
2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.D0*V6)*F2      COUT0454
3D0*U5+128.D0*U6)*F3            COUT0455
48.D0*U5+128.D0*U6)*F3            COUT0456
5+192.D0*V4-48.D0*V5+128.D0*V6)*G3   COUT0457
TM$(3,4)=TM$(3,4)                COUT0458
1-(-4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2      COUT0459
2-(-4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6)*F2      COUT0460
3)*G2                             COUT0461
4.D0*U5-16.D0*U6)*F3            COUT0462
5+48.D0*V4-16.D0*V5-16.D0*V6)*G3   COUT0463
TM$(4,4)=TM$(4,4)                COUT0464
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2      COUT0465
2-(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.D0*V6)*F2      COUT0466
3D0*U5+128.D0*U6)*F3            COUT0467
42.D0*U5+128.D0*U6)*F3            COUT0468
5+384.D0*V4-32.D0*V5+128.D0*V6)*G3   COUT0469
TM$(5,4)=TM$(5,4)                COUT0470

```

```

1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2      COUT0469
2-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F3      COUT0470
3*V6)*G2      COUT0471
4*U5-16.D0*U6)*F3      COUT0472
5.TM$(6,4)=TM$(6,4)      COUT0473
1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F2      COUT0474
2-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F3      COUT0475
3D0*V6)*G2      COUT0476
46.D0*U5+128.D0*U6)*F3      COUT0477
5+128.D0*U4-16.D0*U5+48.D0*U6)*F3      COUT0478
TM$(1,5)=TM$(1,5)      COUT0479
1-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3      COUT0480
2-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3      COUT0481
36)*G3      COUT0482
TM$(2,5)=TM$(2,5)      COUT0483
1-(-16.D0*U1-96.D0*U2-16.D0*U3+16.D0*U4+12.D0*U5+16.D0*U6)*F3      COUT0484
2-(-16.D0*U1-96.D0*U2-16.D0*U3+16.D0*U4+12.D0*U5+16.D0*U6)*F3      COUT0485
3V6)*G3      COUT0486
TM$(3,5)=TM$(3,5)      COUT0487
1-(-11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3      COUT0488
2-(-11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3      COUT0489
3)*G3      COUT0490
TM$(4,5)=TM$(4,5)      COUT0491
1-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3      COUT0492
2-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3      COUT0493
3*V6)*G3      COUT0494
TM$(5,5)=TM$(5,5)      COUT0495
1-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3      COUT0496
2-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3      COUT0497
3)*G3      COUT0498
TM$(6,5)=TM$(6,5)      COUT0499
1-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3      COUT0500
2-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3      COUT0501
3*V6)*G3      COUT0502
TM$(1,6)=TM$(1,6)      COUT0503
1-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1      COUT0504
2-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1      COUT0505
3)*G1      COUT0506
4D0*U5+48.D0*U6)*F3      COUT0507
516.D0*U4-16.D0*U5+48.D0*U6)*G3      COUT0508
TM$(2,6)=TM$(2,6)      COUT0509
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1      COUT0510
2-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1      COUT0511
3D0*V6)*G1      COUT0512
4.D0*U5+192.D0*U6)*F3      COUT0513
5128.D0*U4-48.D0*U5+192.D0*U6)*G3      COUT0514
TM$(3,6)=TM$(3,6)      COUT0515

```

CC

```

1- (4. D0*U1-16. D0*U2+24. D0*U3-32. D0*U4-16. D0*U5-48. D0*U6)*F1
2- (4. D0*V1-16. D0*V2+24. D0*V3-32. D0*V4-16. D0*V5-48. D0*V6)*F1
3) *G1
40*U5-48. D0*U6)*F3
516. D0*V4+4. D0*V5-48. D0*V6)*G3
TM$(4,6)=TM$(4,6)
1- (-48. D0*U1+128. D0*U2-32. D0*U3+384. D0*U4+48. D0*U5+192. D0*U6)*F1
230*V6)*G1
46. D0*U5+128. D0*U6)*F3
5128. D0*V4-16. D0*V5+128. D0*V6)*G3
TM$(5,6)=TM$(5,6)
1- (-16. D0*U1-16. D0*U2-16. D0*U3+48. D0*U4+120. D0*U5+48. D0*U6)*F1
23*V6)*G1
40*U5-32. D0*U6)*F3
56. D0*V4+24. D0*V5-32. D0*V6)*G3
TM$(6,6)=TM$(6,6)
1- (-32. D0*U1+128. D0*U2-48. D0*U3+192. D0*U4+48. D0*U5+384. D0*U6)*F1
230*V6)*G1
40*U5+384. D0*U6)*F3
5128. D0*V4-32. D0*V5+384. D0*V6)*G3

```

THIS ENDS ADDITION OF NONLINEAR TERMS TO THE LOCAL ARRAY

```

TM$(7,7) = TM$(1,1)
TM$(7,8) = TM$(1,2)
TM$(7,9) = TM$(1,3)
TM$(7,10) = TM$(1,4)
TM$(7,11) = TM$(1,5)
TM$(7,12) = TM$(1,6)
TM$(8,7) = TM$(2,1)
TM$(8,8) = TM$(2,2)
TM$(8,9) = TM$(2,3)
TM$(8,10) = TM$(2,4)
TM$(8,11) = TM$(2,5)
TM$(8,12) = TM$(2,6)
TM$(9,7) = TM$(3,1)
TM$(9,8) = TM$(3,2)
TM$(9,9) = TM$(3,3)
TM$(9,10) = TM$(3,4)
TM$(9,11) = TM$(3,5)
TM$(9,12) = TM$(3,6)
TM$(10,7) = TM$(4,1)
TM$(10,8) = TM$(4,2)
TM$(10,9) = TM$(4,3)
TM$(10,10) = TM$(4,4)

```

```

COUT0517
COUT0518
COUT0519
COUT0520
COUT0521
COUT0522
COUT0523
COUT0524
COUT0525
COUT0526
COUT0527
COUT0528
COUT0529
COUT0530
COUT0531
COUT0532
COUT0533
COUT0534
COUT0535
COUT0536
COUT0537
COUT0538
COUT0539
COUT0540
COUT0541
COUT0542
COUT0543
COUT0544
COUT0545
COUT0546
COUT0547
COUT0548
COUT0549
COUT0550
COUT0551
COUT0552
COUT0553
COUT0554
COUT0555
COUT0556
COUT0557
COUT0558
COUT0559
COUT0560
COUT0561
COUT0562
COUT0563
COUT0564

```



```

N(10)=N10
N(11)=N11
N(12)=N12
N(13)=N13
N(14)=N14
N(15)=N15
C
DO 3100 I$=1,15
I = N(I$)
C
DO 3100 J$=1,15
J = N(J$)
TM(I,J) = TM(I,J)+TM$(I$,J$)
3100 CONTINUE
C
3200 CONTINUE
C
DO 3300 I=1,NNQXY
RHS(NQS(I)) = RHS(NQS(I))+Q(NQS(I))
RHS(NQS(I)+NN) = RHS(NQS(I)+NN)+Q(NQS(I)+NN)
3300 CONTINUE
C
3400 CONTINUE
C
INSERT SYSTEM BOUNDARY CONDITIONS
C
DO 3500 I=1,MM
C
DO 3500 J=1,NTOTVP
JX = NV IS(J)
RHS(I) = RHS(I)-TM(I,JX)*T(JX)
TM(I,JX) = 0.00
TM(JX,I) = 0.00
TM(JX,JX) = 1.00
RHS(JX) = T(JX)
3500 CONTINUE
C
M = 1
ND = 82
IDGT = 82
CALL LEQ2F (TM,M,ND,IA,RHS,IDGT,WKAREA,IER)
WRITE (NWRITE,5700)
DO 322 J=1,MM
TDIFF=DABS(T1(J)-T(J))
EPSLN=1.D-06
IF(TDIFF-EPSLN) 322,324,324

```

```

322 CONTINUE
324 CONTINUE
C
DO 3600 I=1,MM
  T(I)=RHS(I)
  T1(I)=T(I)
  WRITE(NWRITE,5800) I,T(I)
CONTINUE
3600 IF(J.NE.MM) GO TO 2840
      WRITE(NWRITE,1091)
C
1090 FORMAT(///,5X,'NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35;
1///,5X,'THE V-VELOCITY AT NODES 36 - 70;
2///,5X,'AND THE PRESSURES AT NODES 71 - 82.',///)
1091 FORMAT(1H1)
1092 FORMAT(///,5X,'THE FIRST SEQUENCE OF 82 NODAL VARIABLES',
1///,5X,'REPRESENTS A LINEAR STEADY STATE SYSTEM;
2///,5X,'WHILE THE SECOND ANALYSIS OF THE 82 VALUES CORRESPONDS',
3///,5X,'TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN.',///)
1093 FORMAT(///,5X,'A PRESSURE GRADIENT IN THE HORIZONTAL SHEAR',
1///,5X,'DIRECTION OF -3 UNITS IN MAGNITUDE HAS BEEN ADDED',
2///,5X,'TO PRODUCE A CURVED VELOCITY PROFILE.',///)
1094 FORMAT(1H1,15X,'THIS IS A 2-D NONLINEAR COUETTE FLOW PROBLEM',//)
3700 FORMAT(5X,'IBAND=',I3,'5X','NEQ =',I3,//)
3800 FORMAT(6X,A4,I10,2F10.0)
3900 FORMAT(6X,A4,I10,F10.0)
4000 FORMAT(5X,'NO. OF NODES=',I3,///)
4100 FORMAT(5X,'NO. OF CORNERS=',I3,///)
4200 FORMAT(5X,'NNVELS=',I3,///)
4300 FORMAT(5X,'NNQXS=',I3,///)
4400 FORMAT(5X,'NNQXS=',I3,///)
4500 FORMAT(5X,'NNQXS=',I3,///)
4600 FORMAT(5X,'NNQXS=',I3,///)
4700 FORMAT(5X,'SUMMARY OF NODAL COORDINATES',//,
17X,I1,I12X,X(I),I13X,Y(I),//)
4800 FORMAT(5X,I3,2(7X,F10.3))
4900 FORMAT(///,5X,'LISTING OF SYSTEM TOPOLOGY',//,5X,
1'ELEMENT NUMBER',20X,'NODE NUMBERS',//)
5000 FORMAT(5X,I3,10X,6(5X,I3))
5100 FORMAT(///,7X,'NODES WHERE U VELOCITY IS SPECIFIED',//)
5200 FORMAT(1,5X,'NODES WHERE V VELOCITY IS SPECIFIED',//)
5300 FORMAT(2X,2(4X,I3),3X,F12.3)
5400 FORMAT(///,5X,'NODES WHERE QX AND QY ARE SPECIFIED',
1///,2X,'NODE',1X,'QX',10X,'QY',//)
5500 FORMAT(///,5X,'NODES WHERE PRESSURE IS SPECIFIED',
1///,5X,'NODE',1X,'PRESSURE',//)
5600 FORMAT(7X,I3,3X,I3,10X,F12.3)
5700 FORMAT(///,5X,'NODE NO.',6X,'NODE VARIABLES',//)

```

```

COUT0709
COUT0710
COUT0711
COUT0712
COUT0713
COUT0714
COUT0715
COUT0716
COUT0717
COUT0718
COUT0719
COUT0720
COUT0721
COUT0722
COUT0723
COUT0724
COUT0725
COUT0726
COUT0727
COUT0728
COUT0729
COUT0730
COUT0731
COUT0732
COUT0733
COUT0734
COUT0735
COUT0736
COUT0737
COUT0738
COUT0739
COUT0740
COUT0741
COUT0742
COUT0743
COUT0744
COUT0745
COUT0746
COUT0747
COUT0748
COUT0749
COUT0750
COUT0751
COUT0752
COUT0753
COUT0754
COUT0755
COUT0756

```

5800 FORMAT (9X,I3,5X,D17.10,/)
STOP
END

COUT0757
COUT0758
COUT0759

STEADY STATE FLUID MECHANICS PROBLEM

```

IMPLICIT REAL*8(A-H,O-Z,$)
DATA NREAD/5/

THE U VELOCITY IS IN THE FIRST NN POSITIONS OF X(I)
THE V VELOCITY IS IN THE SECOND NN POSITIONS OF X I.E. X(I+NN)
THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF X I.E. X(I+NN+NN)
THE T TEMPERATURE IS IN THE NN+NN+NNCN+I POSITIONS OF X I.E.
X(I+NN+NN+NNCN)
THERE ARE NNCN PRESSURE NODES (NNCN=NUMBER OF CORNER NODES)

TM MUST BE DIMENSIONED 3*NN+NNCN X 3*NN+NNCN

DATA NWRITE/6/
DATA STOP/.STOP./
DIMENSION XC(125),YC(125),NODE(125,6),NVS(125),NCN(125)
DIMENSION X(117),NVIS(117),NCP(125),NPS(125),Q(117)
DIMENSION NQS(125),T1(117)
DIMENSION TM$(21,21),N(21),NQIS(117)
DIMENSION RP$(6),ZP$(6),WKAREA(15000)
DIMENSION XC$(3),YC$(3),RHS(117),TM(117,117)

SPECIFY WHETHER TWO DIMENSIONAL, INCLUDING NON-LINEAR TERMS,
(NCASE=1) OR AXISYMMETRIC (NCASE=2)

READ(NREAD,500)NCASE
WRITE(NWRITE,600)
IF(NCASE.EQ.1)GO TO 5
WRITE(NWRITE,2015)
GC TO 6
5 WRITE(NWRITE,2020)
6 CONTINUE

THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE
IN WHICH LINES 540 TO 2970 ARE INPUT VERIFICATION OF ALL DATA.
SUCH A SECTION WOULD BE PART OF ANY FINITE ELEMENT PROGRAM.

```

FLSS0110
 FLSS01120
 FLSS01130
 FLSS01140
 FLSS01150
 FLSS01160
 FLSS01170
 FLSS01180
 FLSS01190
 FLSS01200
 FLSS02210
 FLSS02220
 FLSS02230
 FLSS02240
 FLSS02250
 FLSS02260
 FLSS02270
 FLSS02280
 FLSS02290
 FLSS02300
 FLSS02310
 FLSS02320
 FLSS02330
 FLSS02340
 FLSS02350
 FLSS02360
 FLSS02370
 FLSS02380
 FLSS02390
 FLSS02400
 FLSS02410
 FLSS02420
 FLSS02430
 FLSS02440
 FLSS02460

READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CCRNER NODES

READ(NREAD,1005)NN,NE,NNCN

INITIALIZE ALL PARAMETERS

MM=2*NN+NNCN
MMM=3*NN+NNCN
DO 50 I=1,MMM

XC(I)=0.00
YC(I)=0.00

NVS(I)=0

NCN(I)=0

NCP(I)=0

NPS(I)=0

NGS(I)=0

DO 50 J=1,6

NODE(I,J)=0

CCNTINUE

DO 51 I=1,MMM

TI(I)=0.00

X(I)=0.00

NVIS(I)=0

QC(I)=0.00

RHS(I)=0.00

DO 51 J=1,MMM

TM(I,J)=0.00

CCNTINUE

DO 52 I=1,21

N(I)=0

DO 52 J=1,21

TM\$(I,J)=0.00

CCNTINUE

DO 53 I=1,6

RP\$(I)=0.00

ZP\$(I)=0.00

CCNTINUE

DO 54 I=1,3

XC\$(I)=0.00

YC\$(I)=0.00

CCNTINUE

READ NODE NUMBER AND COORDINATES

DO 100 J=1,NN

FLSS0470
FLSS0480
FLSS0490
FLSS0500
FLSS0510
FLSS0520
FLSS0530
FLSS0540
FLSS0550
FLSS0560
FLSS0570
FLSS0580
FLSS0590
FLSS0600
FLSS0610
FLSS0620
FLSS0630
FLSS0640
FLSS0650
FLSS0660
FLSS0670
FLSS0680
FLSS0690
FLSS0700
FLSS0710
FLSS0720
FLSS0730
FLSS0740
FLSS0750
FLSS0760
FLSS0770
FLSS0780
FLSS0790
FLSS0800
FLSS0810
FLSS0820
FLSS0830
FLSS0840
FLSS0850
FLSS0860
FLSS0870
FLSS0880
FLSS0890
FLSS0900
FLSS0910
FLSS0920
FLSS0930

```

      READ(NREAD,1006)WORD,I,XC(I),YC(I)
      IF(WORD.EQ.STOP) GO TO 101
      NCN(J)=I
      100 CONTINUE
      101 NNCN=J-1

      THE ARRAY NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2,ETC.)
      THUS PRESSURE NODES ARE LABELED AS CORNER NODES AND ARE INPUTED
      WHEN ONE INPUTS A GLOBAL CORNER NODE FOR J

      DC 107 J=1,NNCN
      NCP(NCN(J))=J+NN+NN
      107 CCNTINUE

      READ SYSTEM TOPOLOGY (ELEMENT NO. AND NODE NUMBERS IN
      COUNTERCLOCKWISE FASHION STARTING AT THE UPPER LEFT
      HAND CORNER NODE).

      DO 105 I=1,NE
      READ(NREAD,1010)J,NODE(J,1),NODE(J,2),NODE(J,3),
      1 NODE(J,4),NODE(J,5),NODE(J,6)
      105 CCNTINUE
      MAXDIF=0
      DO 108 I=1,NE
      DC 108 J=1,6
      DO 108 K=1,6
      LL=IABS(NODE(I,J)-NODE(I,K))
      IF(LL.GT.MAXDIF) MAXDIF=LL
      IBAND=2*(MAXDIF+1)
      NEQ=3*NN+NNCN
      108 CONTINUE
      WRITE(NWRITE,1017)IBAND,NEQ
      1017 FORMAT(5X,'IBAND=',I3,'/',5X,'NEQ=',I3,'/')

      READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED

      DO 110 I=1,MM
      READ(NREAD,1006)WORD,NVELS,VELU,VELV
      IF(WORD.EQ.STOP) GO TO 111
      NVS(I)=NVELS
      X(NVS(I))=VELU
      X(NVS(I)+NN)=VELV
      110 CONTINUE

      COUNT NODES HAVING SPECIFIED VELOCITIES

      111 NNVELS=I-1

```

```

FLSS0940
FLSS0950
FLSS0960
FLSS0970
FLSS0980
FLSS0990
FLSS1000
FLSS1010
FLSS1020
FLSS1030
FLSS1040
FLSS1050
FLSS1060
FLSS1070
FLSS1080
FLSS1090
FLSS1100
FLSS1110
FLSS1120
FLSS1130
FLSS1140
FLSS1150
FLSS1160
FLSS1170
FLSS1180
FLSS1190
FLSS1200
FLSS1210
FLSS1220
FLSS1230
FLSS1240
FLSS1250
FLSS1270
FLSS1280
FLSS1290
FLSS1300
FLSS1310
FLSS1320
FLSS1330
FLSS1340
FLSS1350
FLSS1360
FLSS1370
FLSS1380
FLSS1390
FLSS1400
FLSS1410

```

```

READ QX AND QY VALUES AT INTERNAL NODES
DO 125 I=1,NN
  READ(NREAD,1006)WORD,NQXY,QXNS,QYNS
  IF(WORD.EQ.STOP) GO TO 126
  NQS(I)=NQXY
  Q(NQS(I))=QXNS
  Q(NQS(I)+NN)=QYNS
125 CONTINUE
COUNT NODES HAVING SPECIFIED QX AND QY
126 NNQXY=I-1
READ NODE NUMBER AND PRESSURE WHERE SPECIFIED
DO 130 I=1,NN
  READ(NREAD,1025)WORD,NP,PNP
  IF(WORD.EQ.STOP)GO TO 135
  NPS(I)=NP
  X(NCP(NPS(I)))=PNP
130 CONTINUE
COUNT BOUNDARY NODES WHERE PRESSURE IS SPECIFIED
135 NNPS=I-1
READ NODE NUMBER AND TEMPERATURE WHERE SPECIFIED
DO 140 I=1,MM
  READ(NREAD,1025) WORD,NTEMP,TNT
  IF(WORD.EQ.STOP) GO TO 145
  NVS(I+NNVELS)=NTEMP
  X(NVS(I+NNVELS)+MM)=TNT
140 CONTINUE
COUNT NODES HAVING SPECIFIED TEMPERATURES
145 NNTS=I-1
READ NODE NUMBERS AND QZC WHERE SPECIFIED
DO 141 I=1,MM
  READ(NREAD,1025) WORD,NQZC,QZCNS
  IF(WORD.EQ.STOP) GO TO 146
  NQS(NNQXY+I)=NQZC
  Q(NQS(NNQXY+I)+2*NN)=QZCNS
141 CONTINUE

```

```

FLSSS1420
FLSSS1430
FLSSS1440
FLSSS1450
FLSSS1460
FLSSS1470
FLSSS1480
FLSSS1490
FLSSS1500
FLSSS1510
FLSSS1520
FLSSS1530
FLSSS1540
FLSSS1550
FLSSS1560
FLSSS1570
FLSSS1580
FLSSS1590
FLSSS1600
FLSSS1610
FLSSS1620
FLSSS1630
FLSSS1640
FLSSS1650
FLSSS1660
FLSSS1670
FLSSS1680
FLSSS1690
FLSSS1700
FLSSS1710
FLSSS1720
FLSSS1730
FLSSS1740
FLSSS1750
FLSSS1760
FLSSS1770
FLSSS1780
FLSSS1790
FLSSS1800
FLSSS1810
FLSSS1820
FLSSS1830
FLSSS1840
FLSSS1850
FLSSS1860
FLSSS1870
FLSSS1880
FLSSS1890

```

FLSS1900
FLSS1910
FLSS1920
FLSS1930
FLSS1940
FLSS1950
FLSS1960
FLSS1970
FLSS1980
FLSS1990
FLSS2000
FLSS2010
FLSS2020
FLSS2030
FLSS2040
FLSS2050
FLSS2060
FLSS2070
FLSS2080
FLSS2090
FLSS2100
FLSS2110
FLSS2120
FLSS2130
FLSS2140
FLSS2150
FLSS2160
FLSS2170
FLSS2180
FLSS2190
FLSS2200
FLSS2210
FLSS2220
FLSS2230
FLSS2240
FLSS2250
FLSS2260
FLSS2270
FLSS2280
FLSS2290
FLSS2300
FLSS2310
FLSS2320
FLSS2330
FLSS2340
FLSS2350
FLSS2360
FLSS2370

COUNT NODES WHERE QZC IS SPECIFIED

146 NNQZC=I-1

READ NODE NUMBERS AND HEAT FLUX QZ WHERE SPECIFIED

DO 142 I=1,MM
READ(NREAD,1025) WORD,NQZ,QZNS
IF(WORD.EQ.STOP) GO TO 147
NQS(NNQXY+NNQZC+I)=NQZ
Q(NQS(NNQXY+NNQZC+I)+MM)=QZNS
142 CONTINUE

COUNT NODES WHERE HEAT FLUX QZ IS SPECIFIED

147 NNQZ=I-1

NQIS IS A LIST OF INDICES OF KNOWN QX,QY,QZC AND QZ

DO 1140 I=1,NNQXY
NQIS(I)=NQS(I)
NCIS(I+NNQXY)=NQS(I)+NN
1140 CONTINUE

DO 1141 I=1,NNQZC
NQIS(2*NNQXY+I)=NQS(NNQXY+I)+2*NN
1141 CONTINUE

DO 1145 I=1,NNQZ
NQIS(2*NNQXY+NNQZC+I)=NQS(NNQXY+NNQZC+I)+MM
1145 CCNTINUE

NVIS IS A LIST OF KNOWN VELOCITY, PRESSURE, AND TEMPERATURE INDICES

DO 1150 I=1,NNVELS
NVIS(I)=NVS(I)
NVIS(I+NNVELS)=NVS(I)+NN
1150 CONTINUE

DO 1155 J=1,NNPS
NVIS(2*NNVELS+J)=NCP(NPS(J))
1155 CONTINUE

DO 1160 K=1,NNTS
NVIS(2*NNVELS+NNPS+K)=NVS(K+NNVELS)+MM
1160 CONTINUE

NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED

NNHC=0

```

NTOTC=TOTAL NUMBER OF KNOWN QX,QY,QZC, AND QZ
      NTOTQ=2*NNQXY+NNQZC+NNQZ
NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, PRESSURES, AND TEMPERATURES
      NTOTVP=2*NNVELS+NNPS+NNTS

PRINT ALL INPUT DATA
      WRITE(NWRITE,1035)NN,NE,NNCN
      WRITE(NWRITE,1036)NNVELS
      WRITE(NWRITE,1037)NNQXY
      WRITE(NWRITE,1038)NNPS
      WRITE(NWRITE,1039)NNTS
      WRITE(NWRITE,1034)NNQZC
      WRITE(NWRITE,1040)NNQZ
      WRITE(NWRITE,1041)
      DC 150 I=1,NNCN
      WRITE(NWRITE,1045)NCN(I),XC(NCN(I)),YC(NCN(I))
      CONTINUE
      WRITE(NWRITE,1050)
      DO 155 I=1,NE
      WRITE(NWRITE,1055)I,NODE(I,1),NODE(I,2),NODE(I,3),
155 CONTINUE
      WRITE(NWRITE,1060)
      DO 160 I=1,NNVELS
      WRITE(NWRITE,1065)I,NVS(I),X(NVS(I)),X(NVS(I)+NN)
      CONTINUE
      WRITE(NWRITE,1070)
      DO 165 I=1,NNQXY
      WRITE(NWRITE,1065)I,NQS(I),Q(NQS(I)),Q(NQS(I)+NN)
      CONTINUE
      WRITE(NWRITE,1080)
      DO 170 I=1,NNPS
      WRITE(NWRITE,1085)I,NPS(I),X(NCP(NPS(I)))
      CONTINUE
      WRITE(NWRITE,1081)
      DO 171 I=1,NNTS
      WRITE(NWRITE,1085)I,NVS(I+NNVELS),X(NVS(I+NNVELS)+MM)
      CONTINUE
      WRITE(NWRITE,1083)
      DO 173 I=1,NNQZC
      WRITE(NWRITE,1085)I,NQS(I+NNQXY),Q(NQS(I+NNQXY)+2*NN)
      CONTINUE
      WRITE(NWRITE,1082)
      DC 172 I=1,NNQZ

```

```

FLSS2380
FLSS2390
FLSS2400
FLSS2410
FLSS2420
FLSS2430
FLSS2440
FLSS2450
FLSS2460
FLSS2470
FLSS2480
FLSS2490
FLSS2500
FLSS2510
FLSS2520
FLSS2530
FLSS2540
FLSS2550
FLSS2560
FLSS2570
FLSS2580
FLSS2590
FLSS2600
FLSS2610
FLSS2620
FLSS2630
FLSS2640
FLSS2650
FLSS2660
FLSS2670
FLSS2680
FLSS2690
FLSS2700
FLSS2710
FLSS2720
FLSS2730
FLSS2740
FLSS2750
FLSS2760
FLSS2770
FLSS2780
FLSS2790
FLSS2800
FLSS2810
FLSS2820
FLSS2830
FLSS2840
FLSS2850

```

```

      WRITE(NWRITE,1085)I,NQS(I+NNQXY+NNQZC),Q(NQS(I+NNQXY+NNQZC)+MM)
      CONTINUE
172  WRITE(NWRITE,1090)
      WRITE(NWRITE,2055)
      WRITE(NWRITE,1091)
177  CONTINUE
      DO 178 I=1,MMM
      RHS(I)=0.D0
      DO 178 J=1,MMM
      TM(I,J)=0.D0
178  CONTINUE
      DO 181 I=1,21
      DO 181 J=1,21
      TM$(I,J)=0.D0
181  CONTINUE

      END OF INPUT AND VERIFICATION ROUTINE

      DO 300 K=1,NE
      N1=NODE(K,1)
      N2=NODE(K,2)
      N3=NODE(K,3)
      N4=NODE(K,4)
      N5=NODE(K,5)
      N6=NODE(K,6)
      N7=NODE(K,1)+NN
      N8=NODE(K,2)+NN
      N9=NODE(K,3)+NN
      N10=NODE(K,4)+NN
      N11=NODE(K,5)+NN
      N12=NODE(K,6)+NN
      N13=NCP(NODE(K,1))
      N14=NCP(NODE(K,3))
      N15=NCP(NODE(K,5))
      N16=NODE(K,1)+MM
      N17=NODE(K,2)+MM
      N18=NODE(K,3)+MM
      N19=NODE(K,4)+MM
      N20=NODE(K,5)+MM
      N21=NODE(K,6)+MM
      XC$(1)=XC(NODE(K,1))
      XC$(2)=XC(NODE(K,3))
      XC$(3)=XC(NODE(K,5))
      YC$(1)=YC(NODE(K,1))
      YC$(2)=YC(NODE(K,3))
      YC$(3)=YC(NODE(K,5))
      A=1.D0
      ZBAR=(YC$(1)+YC$(2)+YC$(3))/3.D0

```

```

FLSS2860
FLSS2870
FLSS8320
FLSS2880
FLSS2890
FLSS2900
FLSS2910
FLSS2920
FLSS2930
FLSS2940
FLSS2950
FLSS2960
FLSS2970
FLSS2980
FLSS2990
FLSS3000
FLSS3010
FLSS3020
FLSS3030
FLSS3040
FLSS3050
FLSS3060
FLSS3070
FLSS3080
FLSS3090
FLSS3100
FLSS3110
FLSS3120
FLSS3130
FLSS3140
FLSS3150
FLSS3160
FLSS3170
FLSS3180
FLSS3190
FLSS3200
FLSS3210
FLSS3220
FLSS3230
FLSS3240
FLSS3250
FLSS3260
FLSS3270
FLSS3280
FLSS3290
FLSS3300

```

```

RBAR=(XC$(1)+XC$(2)+XC$(3))/3.00
IF(NC=EQ.2)A=RBAR
AA=1.00
IF(NC=EQ.2)AA=2.00*3.1415900*(RBAR)
A1=XC$(2)*YC$(3)-XC$(3)*YC$(2)
A2=XC$(1)*YC$(3)-XC$(3)*YC$(1)
A3=XC$(1)*YC$(2)-XC$(2)*YC$(1)
B1=XC$(2)*YC$(3)-YC$(3)*YC$(1)
B2=XC$(1)*YC$(3)-YC$(3)*YC$(2)
B3=XC$(1)*YC$(2)-YC$(2)*YC$(1)
C1=XC$(1)*XC$(2)-XC$(2)*XC$(3)
C2=XC$(1)*XC$(3)-XC$(3)*XC$(2)
C3=XC$(2)*XC$(3)-XC$(3)*XC$(1)
DEL=DABS(0.500*(XC$(1)*YC$(2)-YC$(3)+XC$(2)*YC$(3)-YC$(1))
1+XC$(3)*YC$(1)-YC$(2))
CONST=10.9000*(1.00*A/(3.00*DEL))*AA
D1=-B1/6.00
D2=-B2/6.00
D3=-B3/6.00
E1=-C1/6.00
E2=-C2/6.00
E3=-C3/6.00
F1=B1/2520.00
F2=B2/2520.00
F3=B3/2520.00
G1=C1/2520.00
G2=C2/2520.00
G3=C3/2520.00
U1=T1(N1)
U2=T1(N2)
U3=T1(N3)
U4=T1(N4)
U5=T1(N5)
U6=T1(N6)
V1=T1(N7)
V2=T1(N8)
V3=T1(N9)
V4=T1(N10)
V5=T1(N11)
V6=T1(N12)
TM$(1,1)=0.7500*(B1*B1+C1*C1)*CONST
TM$(1,2)=(B1*B2+C1*C2)*CONST
TM$(1,3)=-TM$(1,2)*0.2500
TM$(1,4)=0.
TM$(1,6)=(B1*B3+C1*C3)*CONST
TM$(1,5)=-TM$(1,6)*0.2500
TM$(1,3)=0.7500*(B2*B2+C2*C2)*CONST
TM$(3,4)=(B2*B3+C2*C3)*CONST

```

```

TM$(3,5)=-TM$(3,4)*0.25D0
TM$(2,1)=TM$(1,2)
TM$(2,3)=TM$(1,2)
TM$(2,2)=8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)
TM$(2,4)=2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(3,3)
TM$(2,5)=0.
TM$(2,6)=TM$(1,6)+2.D0*TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)
TM$(3,1)=TM$(1,3)
TM$(3,2)=TM$(2,3)
TM$(3,3)=0.
TM$(3,5)=75D0*(B3*B3+C3*C3)*CONST
TM$(4,1)=TM$(1,4)
TM$(4,2)=TM$(2,4)
TM$(4,3)=TM$(3,4)
TM$(4,4)=8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)
TM$(4,5)=TM$(1,6)+TM$(3,4)+TM$(3,4)+TM$(3,4)+TM$(5,5)
TM$(4,6)=TM$(3,6)
TM$(5,1)=TM$(1,5)
TM$(5,2)=TM$(2,5)
TM$(5,3)=TM$(3,5)
TM$(5,4)=TM$(4,5)
TM$(5,5)=TM$(1,6)
TM$(6,1)=TM$(2,6)
TM$(6,2)=TM$(3,6)
TM$(6,3)=TM$(4,6)
TM$(6,4)=TM$(5,6)
TM$(6,5)=8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)
TM$(6,6)=8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)
IF(NC.ASE.NE.1) GO TO 3000

      BEGIN INPUT OF NON-LINEAR TERMS

      TM$(1,1)=TM$(1,1)
      1- (78.D0*U1+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1
      2- (78.D0*V1+48.D0*V2-9.D0*V3+12.D0*V4-9.D0*V5+48.D0*V6)
      3*G1
      TM$(2,1)=TM$(2,1)
      1- (48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1
      2- (48.D0*V1+160.D0*V2-32.D0*V3+16.D0*V4-20.D0*V5+80.D0*
      3*V6)*G1
      TM$(3,1)=TM$(3,1)
      1- (-9.D0*U1-32.D0*U2-18.D0*U3-16.D0*U4+11.D0*U5-20.D0*U6)*F1
      2- (-9.D0*V1-32.D0*V2-18.D0*V3-16.D0*V4+11.D0*V5-20.D0*V6)*G1
      TM$(4,1)=TM$(4,1)
      1- (12.D0*U1+16.D0*U2-16.D0*U3-96.D0*U4-16.D0*U5+16.D0*U6)*F1
      2- (12.D0*V1+16.D0*V2-16.D0*V3-96.D0*V4-16.D0*V5+16.D0*V6)*G1

```

```

FLSSSS3790
FLSSSS3800
FLSSSS3810
FLSSSS3820
FLSSSS3830
FLSSSS3840
FLSSSS3850
FLSSSS3860
FLSSSS3870
FLSSSS3880
FLSSSS3890
FLSSSS3900
FLSSSS3910
FLSSSS3920
FLSSSS3930
FLSSSS3940
FLSSSS3950
FLSSSS3960
FLSSSS3970
FLSSSS3980
FLSSSS3990
FLSSSS4000
FLSSSS4010
FLSSSS4020
FLSSSS4030
FLSSSS4040
FLSSSS4050
FLSSSS4060
FLSSSS4070
FLSSSS4080
FLSSSS4090
FLSSSS4100
FLSSSS4110
FLSSSS4120
FLSSSS4130
FLSSSS4140
FLSSSS4150
FLSSSS4160
FLSSSS4170
FLSSSS4180
FLSSSS4190
FLSSSS4200
FLSSSS4210
FLSSSS4220
FLSSSS4230
FLSSSS4240
FLSSSS4250
FLSSSS4260

```

TM\$(5,1)=TM\$(5,1)
 1-(-9.D0*U1-20.D0*U2+11.D0*U3-16.D0*U4-18.D0*U5-32.D0*U6)*F1
 2-(-9.D0*V1-20.D0*V2+11.D0*V3-16.D0*V4-18.D0*V5-32.D0*V6)*G1
 36)*G1
 TM\$(6,1)=TM\$(6,1)
 1-(-48.D0*U1+80.D0*U2-20.D0*U3+16.D0*U4-32.D0*U5+160.D0*U6)*F1
 2-(-48.D0*V1+80.D0*V2-20.D0*V3+16.D0*V4-32.D0*V5+160.D0*V6)*G1
 36)*G1
 TM\$(1,2)=TM\$(1,2)
 1-(-24.D0*U1-32.D0*U2-16.D0*U3-48.D0*U4+4.D0*U5-16.D0*U6)*F1
 2-(-24.D0*V1-32.D0*V2-16.D0*V3-48.D0*V4+4.D0*V5-16.D0*V6)*G1
 36)*G1
 TM\$(2,2)=TM\$(2,2)
 1-(-32.D0*U1+384.D0*U2+48.D0*U3+192.D0*U4-48.D0*U5+128.D0*U6)*F1
 2-(-32.D0*V1+384.D0*V2+48.D0*V3+192.D0*V4-48.D0*V5+128.D0*V6)*G1
 36)*G1
 TM\$(3,2)=TM\$(3,2)
 1-(-16.D0*U1+48.D0*U2+120.D0*U3+48.D0*U4-16.D0*U5-16.D0*U6)*F1
 2-(-16.D0*V1+48.D0*V2+120.D0*V3+48.D0*V4-16.D0*V5-16.D0*V6)*G1
 36)*G1
 TM\$(4,2)=TM\$(4,2)
 1-(-48.D0*U1+192.D0*U2+48.D0*U3+384.D0*U4-32.D0*U5+128.D0*U6)*F1
 2-(-48.D0*V1+192.D0*V2+48.D0*V3+384.D0*V4-32.D0*V5+128.D0*V6)*G1
 36)*G1
 TM\$(5,2)=TM\$(5,2)
 1-(-4.D0*U1-48.D0*U2-16.D0*U3-32.D0*U4+24.D0*U5-16.D0*U6)*F1
 2-(-4.D0*V1-48.D0*V2-16.D0*V3-32.D0*V4+24.D0*V5-16.D0*V6)*G1
 36)*G1
 TM\$(6,2)=TM\$(6,2)
 1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1
 2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.D0*V6)*G1
 36)*G1
 TM\$(1,3)=TM\$(1,3)
 1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2
 2-(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*V6)*G2
 36)*G2

TM\$(2,3)=TM\$(2,3)
 1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2
 2-(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0*V6)*G2
 3*V6)*G2
 TM\$(3,3)=TM\$(3,3)
 1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2
 2-(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6)*G2
 3*G2
 TM\$(4,3)=TM\$(4,3)
 1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2
 2-(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0*V6)*G2
 3*V6)*G2
 TM\$(5,3)=TM\$(5,3)
 1-(-11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2
 2-(-11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6)*G2
 3*G2
 TM\$(6,3)=TM\$(6,3)
 1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2
 2-(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*V6)*G2
 3*V6)*G2
 TM\$(1,4)=TM\$(1,4)
 1-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2
 2-(-24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6)*G2
 3*G2
 TM\$(2,4)=TM\$(2,4)
 1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5-32.D0*U6)*F2
 2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5-32.D0*V6)*G2
 3*G2
 TM\$(3,4)=TM\$(3,4)
 1-(-4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2
 2-(-4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6)*G2
 3*G2
 TM\$(4,4)=TM\$(4,4)
 1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2
 2-(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.D0*V6)*G2
 3*G2
 TM\$(5,4)=TM\$(5,4)
 1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2
 2-(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0*V6)*G2
 3*V6)*G2

FLSS4750
 FLSS4760
 FLSS4770
 FLSS4780
 FLSS4790
 FLSS4800
 FLSS4810
 FLSS4820
 FLSS4830
 FLSS4840
 FLSS4850
 FLSS4860
 FLSS4870
 FLSS4880
 FLSS4890
 FLSS4900
 FLSS4910
 FLSS4920
 FLSS4930
 FLSS4940
 FLSS4950
 FLSS4960
 FLSS4970
 FLSS4980
 FLSS4990
 FLSS5000
 FLSS5010
 FLSS5020
 FLSS5030
 FLSS5040
 FLSS5050
 FLSS5060
 FLSS5070
 FLSS5080
 FLSS5090
 FLSS5100
 FLSS5110
 FLSS5120
 FLSS5130
 FLSS5140
 FLSS5150
 FLSS5160
 FLSS5170
 FLSS5180
 FLSS5190
 FLSS5200
 FLSS5210
 FLSS5220

4*U5-16.00*U6)*F3
 5.00*V4+24.00*V5-16.00*V6)*G3
 1-(-32.00*U1+128.00*U2-48.00*U3+192.00*U4+48.00*U5+384.00*U6)*F2
 2-(-32.00*U1+128.00*U2-48.00*U3+192.00*U4+48.00*U5+384.00*U6)*F2
 300*V6)*G2
 46.00*U5+128.00*U6)*F3
 5+128.00*V4-16.00*V5+128.00*V6)*G3
 1-(-18.00*U1-16.00*U2+11.00*U3-20.00*U4-9.00*U5-32.00*U6)*F3
 2-(-18.00*U1-16.00*U2+11.00*U3-20.00*U4-9.00*U5-32.00*U6)*F3
 36)*G3
 1-(-16.00*U1-96.00*U2-16.00*U3+16.00*U4+12.00*U5+16.00*U6)*F3
 2-(-16.00*U1-96.00*U2-16.00*U3+16.00*U4+12.00*U5+16.00*U6)*F3
 3V6)*G3
 1-(-11.00*U1-16.00*U2-18.00*U3-32.00*U4-9.00*U5-20.00*U6)*F3
 2-(-11.00*U1-16.00*U2-18.00*U3-32.00*U4-9.00*U5-20.00*U6)*F3
 3)*G3
 1-(-20.00*U1+16.00*U2-32.00*U3+160.00*U4+48.00*U5+80.00*U6)*F3
 2-(-20.00*U1+16.00*U2-32.00*U3+160.00*U4+48.00*U5+80.00*U6)*F3
 3*V6)*G3
 1-(-9.00*U1+12.00*U2-9.00*U3+48.00*U4+78.00*U5+48.00*U6)*F3
 2-(-9.00*U1+12.00*U2-9.00*U3+48.00*U4+78.00*U5+48.00*U6)*F3
 3)*G3
 1-(-32.00*U1+16.00*U2-20.00*U3+80.00*U4+48.00*U5+160.00*U6)*F3
 2-(-32.00*U1+16.00*U2-20.00*U3+80.00*U4+48.00*U5+160.00*U6)*F3
 3*V6)*G3
 1-(-24.00*U1-16.00*U2+4.00*U3-48.00*U4-16.00*U5-32.00*U6)*F1
 2-(-24.00*U1-16.00*U2+4.00*U3-48.00*U4-16.00*U5-32.00*U6)*F1
 3)*G1
 40*U5+48.00*U6)*F3
 516.00*V4-16.00*V5+48.00*V6)*G3
 1-(-16.00*U1+128.00*U2-16.00*U3+128.00*U4-16.00*U5+128.00*U6)*F1
 2-(-16.00*U1+128.00*U2-16.00*U3+128.00*U4-16.00*U5+128.00*U6)*F1
 300*V6)*G1
 4.00*U5+192.00*U6)*F3
 5128.00*V4-48.00*V5+192.00*V6)*G3
 1-(-4.00*U1-16.00*U2+24.00*U3-32.00*U4-16.00*U5-48.00*U6)*F1
 2-(-4.00*U1-16.00*U2+24.00*U3-32.00*U4-16.00*U5-48.00*U6)*F1
 3)*G1

```

40*U5-48.D0*U6)*F3
516.D0*V4+4.D0*V5-48.D0*V6)*G3
TM$(4,6)=TM$(4,6)
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1
2-(-48.D0*U1+128.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.D0*V6)*G1
3D0*V6)*G1
46.D0*U5+128.D0*U6)*F3
5+128.D0*V4-16.D0*V5+128.D0*V6)*G3
TM$(5,6)=TM$(5,6)
1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F1
2-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*G1
3*V6)*G1
40*U5-32.D0*U6)*F3
56.D0*V4+24.D0*V5-32.D0*V6)*G3
TM$(6,6)=TM$(6,6)
1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F1
2-(-32.D0*U1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.D0*V6)*G1
3D0*V6)*G1
4.D0*U5+384.D0*U6)*F3
5128.D0*V4-32.D0*V5+384.D0*V6)*G3
-(-16.D0*V1-32.D0*V2+24.D0*V3-
FLSS55710
FLSS55720
FLSS55730
FLSS55740
FLSS55750
FLSS55760
FLSS55770
FLSS55780
FLSS55790
FLSS55800
FLSS55810
FLSS55820
FLSS55830
FLSS55840
FLSS55850
FLSS55860
FLSS55870
FLSS55880
FLSS55890
FLSS55900
FLSS55910
FLSS55920
FLSS55930
FLSS55940
FLSS55950
FLSS55960
FLSS55970
FLSS55980
FLSS55990
FLSS6000
FLSS6010
FLSS6020
FLSS6030
FLSS6040
FLSS6050
FLSS6060
FLSS6070
FLSS6080
FLSS6090
FLSS6100
FLSS6110
FLSS6120
FLSS6130
FLSS6140
FLSS6150
FLSS6160
FLSS6170
FLSS6180

```

THIS ENCS ADDITION OF NON-LINEAR TERMS TO THE LOCAL ARRAY

3000

```

CONTINUE
TM$(7,7)=TM$(1,1)
TM$(7,8)=TM$(1,2)
TM$(7,9)=TM$(1,3)
TM$(7,10)=TM$(1,4)
TM$(7,11)=TM$(1,5)
TM$(7,12)=TM$(1,6)
TM$(8,7)=TM$(2,1)
TM$(8,8)=TM$(2,2)
TM$(8,9)=TM$(2,3)
TM$(8,10)=TM$(2,4)
TM$(8,11)=TM$(2,5)
TM$(8,12)=TM$(2,6)
TM$(9,7)=TM$(3,1)
TM$(9,8)=TM$(3,2)
TM$(9,9)=TM$(3,3)
TM$(9,10)=TM$(3,4)
TM$(9,11)=TM$(3,5)
TM$(9,12)=TM$(3,6)
TM$(10,7)=TM$(4,1)
TM$(10,8)=TM$(4,2)
TM$(10,9)=TM$(4,3)
TM$(10,10)=TM$(4,4)
TM$(10,11)=TM$(4,5)
TM$(10,12)=TM$(4,6)

```

[illegible]

```

CONST4=0.94375D0
CMTM$(1,1,1,3)=D1*CONST4
CMTM$(1,1,1,4)=0.D0
CMTM$(1,1,1,5)=0.D0
CMTM$(2,1,1,3)=(D1+2.D0*D2)*CONST4
CMTM$(2,1,1,4)=(2.D0*D1+D2)*CONST4
CMTM$(2,1,1,5)=(D1+D2)*CONST4
CMTM$(3,1,1,3)=0.D0
CMTM$(3,1,1,4)=D2*CONST4
CMTM$(3,1,1,5)=0.D0
CMTM$(4,1,1,3)=(D2+D3)*CONST4
CMTM$(4,1,1,4)=(D2+2.D0*D3)*CONST4
CMTM$(4,1,1,5)=(2.D0*D2+D3)*CONST4
CMTM$(5,1,1,3)=0.D0
CMTM$(5,1,1,4)=0.D0
CMTM$(5,1,1,5)=D3*CONST4
CMTM$(6,1,1,3)=(D1+2.D0*D3)*CONST4
CMTM$(6,1,1,4)=(D1+D3)*CONST4
CMTM$(6,1,1,5)=(2.D0*D1+D3)*CONST4
CMTM$(7,1,1,3)=E1*CONST4
CMTM$(7,1,1,4)=0.D0
CMTM$(7,1,1,5)=0.D0
CMTM$(8,1,1,3)=(E1+2.D0*E2)*CONST4
CMTM$(8,1,1,4)=(E1+D0*E1+E2)*CONST4
CMTM$(8,1,1,5)=(E1+E2)*CONST4
CMTM$(9,1,1,3)=0.D0
CMTM$(9,1,1,4)=E2*CONST4
CMTM$(9,1,1,5)=0.D0
CMTM$(10,1,1,3)=(E2+E3)*CONST4
CMTM$(10,1,1,4)=(E2+2.D0*E3)*CONST4
CMTM$(10,1,1,5)=(E2+E3)*CONST4
CMTM$(11,1,1,3)=0.D0
CMTM$(11,1,1,4)=0.D0
CMTM$(11,1,1,5)=E3*CONST4
CMTM$(12,1,1,3)=(E1+2.D0*E3)*CONST4
CMTM$(12,1,1,4)=(E1+E3)*CONST4
CMTM$(12,1,1,5)=(2.D0*E1+E3)*CONST4
CMTM$(12,1,1,6)=3.2981D0
CMTM$(1,1,1,6)=1.D0*CONST2
CMTM$(2,1,1,7)=1.D0*CONST2
CMTM$(3,1,1,8)=1.D0*CONST2
CMTM$(4,1,1,9)=1.D0*CONST2
CMTM$(5,1,1,20)=1.D0*CONST2
CMTM$(6,1,1,21)=1.D0*CONST2

```

ALPHA1= ALPHA/ VISCOSITY

ALPHA1=9.4454D-05

```

FLSS6670
FLSS6680
FLSS6690
FLSS6700
FLSS6710
FLSS6720
FLSS6730
FLSS6740
FLSS6750
FLSS6760
FLSS6770
FLSS6780
FLSS6790
FLSS6800
FLSS6810
FLSS6820
FLSS6830
FLSS6840
FLSS6850
FLSS6860
FLSS6870
FLSS6880
FLSS6890
FLSS6900
FLSS6910
FLSS6920
FLSS6930
FLSS6940
FLSS6950
FLSS6960
FLSS6970
FLSS6980
FLSS6990
FLSS7000
FLSS7010
FLSS7020
FLSS7030
FLSS7040
FLSS7050
FLSS7060
FLSS7070
FLSS7080
FLSS7090
FLSS7100
FLSS7110
FLSS7120
FLSS7130
FLSS7140

```



```

N(11)=N11
N(12)=N12
N(13)=N13
N(14)=N14
N(15)=N15
N(16)=N16
N(17)=N17
N(18)=N18
N(19)=N19
N(20)=N20
N(21)=N21
DO 200 I$=1,21
I=N(I$)
DO 200 J$=1,21
J=N(J$)
TM(I,J)=TM(I,J)+TM$(I$,J$)
200 CONTINUE
300 IF(NNQXY.EQ.0) GO TO 310
DO 310 I=1,NNQXY
RHS(NQS(I))=RHS(NQS(I))+Q(NQS(I))+65.962D0
RHS(NQS(I)+NN)=RHS(NQS(I)+NN)+Q(NQS(I)+NN)
310 CONTINUE
310 IF(NNQZC.EQ.0) GO TO 312
DO 312 I=1,NNQZC
RHS(NQS(NNQXY+I)+2*NN)=RHS(NQS(NNQXY+I))+2*NN)+Q(NQS(NNQXY+I))+2*NN)
312 CONTINUE
312 IF(NNQZ.EQ.0) GO TO 311
DO 311 I=1,NNQZ
RHS(NQS(NNQXY+NNQZC+I)+MM)=RHS(NQS(NNQXY+NNQZC+I)+MM)+
1Q(NQS(NNQXY+NNQZC+I)+MM)
311 CONTINUE

MODIFICATION OF RHS FOR TM BOUNDARY CONDITIONS

DO 315 I=1,MMM
DO 315 J=1,NTOTVP
JX=NVIS(J)
RHS(I)=RHS(I)-TM(I,JX)*X(JX)
TM(I,JX)=0.D0
TM(JX,I)=0.D0
TM(JX,JX)=1.D0
RHS(JX)=X(JX)
315 CONTINUE
M=1
ND=117
IA=117
IDGT=0

```

```

FLSSS7630
FLSSS7640
FLSSS7650
FLSSS7660
FLSSS7670
FLSSS7680
FLSSS7690
FLSSS7700
FLSSS7710
FLSSS7720
FLSSS7730
FLSSS7740
FLSSS7750
FLSSS7760
FLSSS7770
FLSSS7780
FLSSS7790
FLSSS7800
FLSSS7810
FLSSS7820
FLSSS7830
FLSSS7840
FLSSS7850
FLSSS7860
FLSSS7870
FLSSS7880
FLSSS7890
FLSSS7900
FLSSS7910
FLSSS7920
FLSSS7930
FLSSS7940
FLSSS7950
FLSSS7960
FLSSS7970
FLSSS7980
FLSSS7990
FLSSS8000
FLSSS8010
FLSSS8020
FLSSS8030
FLSSS8040
FLSSS8050
FLSSS8060
FLSSS8070
FLSSS8080
FLSSS8090
FLSSS8100

```

```

CALL LEQT2F (TM,M,ND,IA,RHS,IDGT,WKAREA,IER)
WRITE(NWRITE,2000)
DO 322 J=1,MMM
TDIFF=DABS(T1(J)-X(J))
EPSLN=1.D-06
IF(TDIFF-EPSLN) 322,324,324
322 CONTINUE
324 CONTINUE
DO 360 I=1,MMM
X(I)=RHS(I)
T1(I)=X(I)
WRITE (NWRITE,2005) I,X(I)
360 CONTINUE
IF(J.NE.MMM) GO TO 177
WRITE(NWRITE,2021)
WRITE(NWRITE,2025)
WRITE(NWRITE,2030)
WRITE(NWRITE,2035)
WRITE(NWRITE,2040)
WRITE(NWRITE,2045)
WRITE(NWRITE,2050)
FORMAT(I10)
500 FORMAT(IH1,18X,'STEADY STATE FLUID MECHANICS PROBLEM',////)
600 FORMAT(3I10)
1005 FORMAT(6X,A4,I10,2F10.0)
1006 FORMAT(7I10)
1010 FORMAT(6X,A4,I10,I10,F10.0)
1015 FORMAT(6X,A4,I10,2F10.0)
1016 FORMAT(6X,A4,I10,2F10.0)
1020 FORMAT(6X,A4,I10,F10.0)
1025 FORMAT(6X,A4,2I10,2F10.0)
1030 FORMAT(5X,'NNQZC=',I3,/)
1034 FORMAT(5X,'NO. OF NODES=',I3,/)
1035 15X,'NO. OF CORNERS=',I3,/)
1036 FORMAT(5X,'NNVELS=',I3,/)
1037 FORMAT(5X,'NNQXY=',I3,/)
1038 FORMAT(5X,'NNPS=',I3,/)
1039 FORMAT(5X,'NNTS=',I3,/)
1040 FORMAT(5X,'NNQZ=',I3,/)
1041 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,
1045 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,
1050 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,
1055 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,
1060 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,
1065 17X,'I.T(5X,I3,2(7X,F10.3)) SUMMARY OF NODAL COORDINATES',/,

```

```

FLSS8110
FLSS8120
FLSS8130
FLSS8140
FLSS8150
FLSS8160
FLSS8170
FLSS8180
FLSS8190
FLSS8200
FLSS8210
FLSS8220
FLSS8230
FLSS8240
FLSS8250
FLSS8260
FLSS8270
FLSS8280
FLSS8290
FLSS8300
FLSS8310
FLSS8330
FLSS8340
FLSS8350
FLSS8360
FLSS8370
FLSS8380
FLSS8390
FLSS8400
FLSS8410
FLSS8420
FLSS8430
FLSS8440
FLSS8450
FLSS8460
FLSS8470
FLSS8480
FLSS8490
FLSS8500
FLSS8510
FLSS8520
FLSS8540
FLSS8550
FLSS8560
FLSS8570
FLSS8580

```

```

1070 FORMAT(//,5X,'NODES WHERE QX AND QY ARE SPECIFIED',
1071         ,2X,'NODE',1X,'QX',10X,'QY',//)
1075 FORMAT(5X,I3,2(10X,F12.3))
1080 FORMAT(//,5X,'NODES WHERE PRESSURE IS SPECIFIED',
1081         ,9X,I',4X,NODE',13X,'PRESSURE',//)
1081 FORMAT(//,5X,'NODES WHERE TEMPERATURE IS SPECIFIED',
1082         ,9X,I',4X,NODE',12X,'TEMPERATURE',//)
1082 FORMAT(//,5X,'NODES WHERE HEAT FLUX QZ IS SPECIFIED',
1083         ,9X,I',4X,NODE',12X,'HEAT FLUX',//)
1083 FORMAT(//,5X,'NODES WHERE QZC IS SPECIFIED',
1084         ,9X,I',4X,NODE',15X,'QZC',//)
1085 FORMAT(7X,I3,3X,I3,10X,F12.3)
1090 FORMAT(//,5X,'NODAL VARIABLE IS THE U-VELOCITY AT NODES 1 - 35',
1091         ,5X,'THE V-VELOCITY AT NODES 36 - 70',
1092         ,5X,'AND THE PRESSURE AT NODES 71 - 82',
1093         ,5X,'THE TEMPERATURE AT NODES 83 - 117',//)
1091 FORMAT(//,5X,'THE FIRST SEQUENCE OF 117 NODAL VARIABLES',
1092         ,5X,'REPRESENTS A LINEAR, STEADY STATE SYSTEM',
1093         ,5X,'WHILE THE SECOND SET OF THE 117 VALUES CORRESPONDS',
1094         ,5X,'TO A NONLINEAR ANALYSIS OF THE SOLUTION DOMAIN',//)
1095 FORMAT(5X,3(3X,I3),2(5X,F12.3))
2000 FORMAT(//,5X,'NODE NO.',6X,'NODE VARIABLE',//)
2005 FORMAT(9X,I3,5X,D17.10,/)
2015 FORMAT(1H1,21X,'THIS IS A 3-D AXISYMMETRIC PROBLEM',//)
2020 FORMAT(21X,'THIS IS A 2-D NONLINEAR PROBLEM',//)
2021 FORMAT(//,5X,10X,'THE KINEMATIC VISCOSITY OF FLUID 50-HB-3520 AT 20
1DEGREES C.',//)
2025 FORMAT(//,5X,10X,'THE DENSITY OF 50-HB-3520 AT 20 DEGREES C. = 1.059',
2026         ,10X,'SQ.CM/SEC',//)
2030 FORMAT(//,5X,10X,'THE COEFF. OF THERMAL EXPANSION OF 50-HB-3520 AT 20
16 GM/CC',//)
2035 FORMAT(//,5X,10X,'THE THERMAL DIFFUSIVITY OF 50-HB-3520 AT 20 DEGREE',
2036         ,10X,'DEGREES C.',//)
2040 FORMAT(//,5X,10X,'THE GRASHOF NUMBER (GR(L)) = (G*B*L**3*(TH-TC))/V**',
2041         ,10X,'SQ.CM/SEC',//)
2045 FORMAT(//,5X,10X,'THE U VELOCITY FORCING FUNCTION, G*B*T(INITIAL), =
12 = 946.4',//)
2050 FORMAT(//,5X,10X,'THE SPECIFIED WALL PRESSURES',
2051         ,165.962 CM/SQ.SEC',//)
2055 FORMAT(//,5X,10X,'THE SPECIFIED WALL PRESSURES',
2056         ,1X,'5X,'ARE NORMALIZED TO ONE (1) ATMOSPHERE',
2057         ,3X,'5X,'THAT IS, 1014000 DYNES/SQ.CM',
2058         ,3X,'5X,'(ALL PARAMETER VALUES ARE IN CGS UNITS)',//)
2059 STOP
2060 END

```

FLSS8590
FLSS8600
FLSS8610
FLSS8620
FLSS8630
FLSS8640
FLSS8650
FLSS8660
FLSS8670
FLSS8680
FLSS8690
FLSS8700

FLSS8710
FLSS8720
FLSS8730
FLSS8740
FLSS8750
FLSS8760
FLSS8770

FLSS8790
FLSS8800
FLSS8810

FLSS8850
FLSS8860
FLSS8870
FLSS8880

FLSS8910
FLSS8920

TIME-DEPENDENT FLUID MECHANICS PROBLEM

```

IMPLICIT REAL*8(A-H,O-Z,$)
DATA NREAD/5/

THE U VELOCITY IS IN THE FIRST NN POSITIONS OF X(I)
THE V VELOCITY IS IN THE SECOND NN POSITIONS OF X I.E. X(I+NN)
THE P PRESSURE IS IN THE NN+NN+I POSITIONS OF X I.E. X(I+NN+NN)
THE T TEMPERATURE IS IN THE NN+NN+NNCN+I POSITIONS OF X I.E.
X(I+NN+NN+NNCN)
THERE ARE NNCN PRESSURE NODES (NNCN=NUMBER OF CORNER NODES)
TM MUST BE DIMENSIONED 3*NN+NNCN X 3*NN+NNCN

DATA NWRITE/6/
DATA STOP/STOP//
DIMENSION XC(125),YC(125),NODE(125,6),NVS(125),NCN(125)
DIMENSION X(117),NVIS(117),NCP(125),NPS(125),Q(117)
DIMENSION TMS(21,21),TI(117)
DIMENSION RPS(6),ZPS(6)
DIMENSION XCS(3),YCS(3)
DIMENSION CDS(21,21)
DIMENSION Y(7,117),W(117,132)
COMMON CD(117,117),TM(117,117),RHS(117)

SPECIFY WHETHER TWO DIMENSIONAL, INCLUDING NON-LINEAR TERMS,
(NCASE=1) OR AXISYMMETRIC (NCASE=2)

      READ(NREAD,500)NCASE
      IF(NCASE.EQ.1)GO TO 5
      WRITE(NWRITE,2015)
      GO TO 6
      5 WRITE(NWRITE,2020)
      6 CONTINUE

TIME00001
TIME00002
TIME00003
TIME00004
TIME00005
TIME00006
TIME00007
TIME00008
TIME00009
TIME00010
TIME00011
TIME00012
TIME00013
TIME00014
TIME00015
TIME00016
TIME00017
TIME00018
TIME00019
TIME00020
TIME00021
TIME00022
TIME00023
TIME00024
TIME00025
TIME00026
TIME00027
TIME00028
TIME00029
TIME00030
TIME00031
TIME00032
TIME00033
TIME00034
TIME00035
TIME00036

```

THE FIRST PART OF THE PROGRAM CAN BE CONSIDERED AS AN INPUT ROUTINE

IN WHICH LINES 049 TO 0303 ARE INPUT VERIFICATION OF ALL DATA.
SUCH A SECTION WOULD BE PART OF ANY FINITE ELEMENT PROGRAM.

READ IN NUMBER OF NODES AND ELEMENTS AND NO. OF CORNER NODES

READ(NREAD,1005)NN,NE,NNCN

INITIALIZE ALL PARAMETERS

```

MM=2*NN+NNCN
MMM=3*NN+NNCN
DO 50 I=1,MMM
  XC(I)=0.00
  YC(I)=0.00
  NVS(I)=0
  NCP(I)=0
  NPS(I)=0
  NQS(I)=0
  DO 50 J=1,6
    NODE(I,J)=0
  CONTINUE
  DO 51 I=1,MMM
    TI(I)=0.00
    Y(I)=0.00
    X(I)=0.00
    NVIS(I)=0
    Q(I)=0.00
    NQIS(I)=0.00
    RHS(I)=0.00
    DO 51 J=1,MMM
      TM(I,J)=0.00
      CC(I,J)=0.00
    CONTINUE
  DO 52 I=1,21
    N(I)=0
    DO 52 J=1,21
      TM$(I,J)=0.00
      CD$(I,J)=0.00
    CONTINUE
  DO 53 I=1,6
    RP$(I)=0.00
    ZP$(I)=0.00
  CONTINUE
  DO 54 I=1,3
    XC$(I)=0.00
    YC$(I)=0.00

```

50

51

52

53

TIME0038
TIME0040
TIME0041
TIME0042
TIME0043
TIME0044
TIME0045
TIME0046
TIME0047
TIME0048
TIME0049
TIME0050
TIME0051
TIME0052
TIME0053
TIME0054
TIME0055
TIME0056
TIME0057
TIME0058
TIME0059
TIME0060
TIME0061
TIME0062
TIME0063
TIME0064
TIME0065
TIME0066
TIME0067
TIME0068
TIME0069
TIME0070
TIME0071
TIME0072
TIME0073
TIME0074
TIME0075
TIME0076
TIME0077
TIME0078
TIME0079
TIME0080
TIME0081
TIME0082
TIME0083
TIME0084
TIME0085
TIME0086

```

54 CONTINUE
READ NODE NUMBER AND COORDINATES
DO 100 J=1,NN
  READ(NREAD,1006)WORD,I,XC(I),YC(I)
  IF(WORD.EQ.STOP) GO TO 101
  NCN(J)=I
  CONTINUE
100 NCN=J-1
  101 NNCN=J-1

THE ARRAY NCP(J) GENERATES THE GLOBAL PRESSURE INDICES (P1,P2,ETC.)
THUS PRESSURE NODES ARE LABELED AS CORNER NODES AND ARE INPUTED
WHEN CNE INPUTS A GLOBAL CORNER NODE FOR J
DO 107 J=1,NNCN
  NCP(NCN(J))=J+NN+NN
107 CONTINUE

READ SYSTEM TOPOLOGY (ELEMENT NO. AND NODE NUMBERS IN
COUNTERCLOCKWISE FASHION STARTING AT THE UPPER LEFT
HAND CORNER NODE).
DO 105 I=1,NE
  READ(NREAD,1010)J,NODE(J,1),NODE(J,2),NODE(J,3),
  1 NODE(J,4),NODE(J,5),NODE(J,6)
105 CONTINUE
  MAXDIF=0
  DO 108 I=1,NE
    DO 108 J=1,6
      DO 108 K=1,6
        LL=IABS(NODE(I,J)-NODE(I,K))
        IF(LL.GT.MAXDIF) MAXDIF=LL
        IBAND=2*(MAXDIF+1)
        NEQ=3*NN+NNCN
      CONTINUE
108 WRITE(NWRITE,1017)IBAND,NEQ
1017 FORMAT(5X,'IBAND=',I3,'/',5X,'NEQ =',I3,/)

READ NODES WHERE BOTH U AND V VELOCITY IS SPECIFIED
DO 110 I=1,MM
  READ(NREAD,1006)WORD,NVELS,VELU,VELV
  IF(WORD.EQ.STOP) GO TO 111
  NVS(I)=NVELS
  X(NVS(I))=VELU
  X(NVS(I))+NN=VELV
110 CONTINUE

```

```

TIME0087
TIME0088
TIME0089
TIME0090
TIME0091
TIME0092
TIME0093
TIME0094
TIME0095
TIME0096
TIME0097
TIME0098
TIME0099
TIME0100
TIME0101
TIME0102
TIME0103
TIME0104
TIME0105
TIME0106
TIME0107
TIME0108
TIME0109
TIME0110
TIME0111
TIME0112
TIME0113
TIME0114
TIME0115
TIME0116
TIME0117
TIME0118
TIME0119
TIME0120
TIME0121
TIME0122
TIME0123
TIME0124
TIME0125
TIME0126
TIME0127
TIME0128
TIME0129
TIME0130
TIME0131
TIME0132
TIME0133
TIME0134

```

TIME0135
TIME0136
TIME0137
TIME0138
TIME0139
TIME0140
TIME0141
TIME0142
TIME0143
TIME0144
TIME0145
TIME0146
TIME0147
TIME0148
TIME0149
TIME0150
TIME0151
TIME0152
TIME0153
TIME0154
TIME0155
TIME0156
TIME0157
TIME0158
TIME0159
TIME0160
TIME0161
TIME0162
TIME0163
TIME0164
TIME0165
TIME0166
TIME0167
TIME0168
TIME0169
TIME0170
TIME0171
TIME0172
TIME0173
TIME0174
TIME0175
TIME0176
TIME0177
TIME0178
TIME0179
TIME0180
TIME0181
TIME0182

COUNT NODES HAVING SPECIFIED VELOCITIES

111 NNVELS=I-1

READ QX AND QY VALUES AT INTERNAL NODES

```
DO 125 I=1,NN
  READ(NREAD,1006)WORD,NQXY,QXNS,QYNS
  IF(WORD.EQ.STOP) GO TO 126
  NQS(I)=NQXY
  Q(NQS(I))=QXNS
  Q(NQS(I)+NN)=QYNS
125 CONTINUE
```

COUNT NODES HAVING SPECIFIED QX AND QY

126 NNQXY=I-1

READ NOCE NUMBER AND PRESSURE WHERE SPECIFIED

```
DO 130 I=1,NN
  READ(NREAD,1025)WORD,NP,PNP
  IF(WORD.EQ.STOP)GO TO 135
  NPS(I)=NP
  X(NCP(NPS(I)))=PNP
130 CONTINUE
```

COUNT BOUNDARY NODES WHERE PRESSURE IS SPECIFIED

135 NNPS=I-1

READ NOCE NUMBER AND TEMPERATURE WHERE SPECIFIED

```
DO 140 I=1,MM
  READ(NREAD,1025) WORD,NTEMP,TNT
  IF(WORD.EQ.STOP) GO TO 145
  NVS(I+NNVELS)=NTEMP
  X(NVS(I+NNVELS)+MM)=TNT
140 CONTINUE
```

COUNT NODES HAVING SPECIFIED TEMPERATURES

145 NNTS=I-1

READ NOCE NUMBERS AND QZC WHERE SPECIFIED

DO 141 I=1,MM

```

      REAC(NREAD,1025) WORD,NQZC,QZCNS
      IF(WORD.EQ.STOP) GO TO 146
      NQS(NNQXY+I)=NQZC
      Q(NQS(NNQXY+I)+2*NN)=QZCNS
141  CONTINUE

```

```

      COUNT NODES WHERE QZC IS SPECIFIED

```

```

146  NNQZC=I-1

```

```

      READ NODE NUMBERS AND HEAT FLUX QZ WHERE SPECIFIED

```

```

      DO 142 I=1,MM
      READ(NREAD,1025) WORD,NQZ,QZNS
      IF(WORD.EQ.STOP) GO TO 147
      NQS(NNQXY+NNQZC+I)=NQZ
      Q(NQS(NNQXY+NNQZC+I)+MM)=QZNS
142  CONTINUE

```

```

      COUNT NODES WHERE HEAT FLUX QZ IS SPECIFIED

```

```

147  NNQZ=I-1

```

```

      NQIS IS A LIST OF INDICES OF KNOWN QX,QY,QZC AND QZ

```

```

      DO 1140 I=1,NNQXY
      NQIS(I)=NQS(I)
      NQIS(I+NNQXY)=NQS(I)+NN
1140  CONTINUE

```

```

      DO 1141 I=1,NNQZC
      NQIS(2*NNQXY+I)=NQS(NNQXY+I)+2*NN
1141  CONTINUE

```

```

      DO 1145 I=1,NNQZ
      NQIS(2*NNQXY+NNQZC+I)=NQS(NNQXY+NNQZC+I)+MM
1145  CONTINUE

```

```

      NVIS IS A LIST OF KNOWN VELOCITY, PRESSURE, AND TEMPERATURE INDICES

```

```

      DO 1150 I=1,NNVELS
      NVIS(I)=NVS(I)
      NVIS(I+NNVELS)=NVS(I)+NN
1150  CONTINUE

```

```

      DO 1155 J=1,NNPS
      NVIS(2*NNVELS+J)=NCP(NPS(J))
1155  CONTINUE

```

```

      DO 1160 K=1,NNTS
      NVIS(2*NNVELS+NNPS+K)=NVS(K+NNVELS)+MM
1160  CONTINUE

```

```

TIME0183
TIME0184
TIME0185
TIME0186
TIME0187
TIME0188
TIME0189
TIME0190
TIME0191
TIME0192
TIME0193
TIME0194
TIME0195
TIME0196
TIME0197
TIME0198
TIME0199
TIME0200
TIME0201
TIME0202
TIME0203
TIME0204
TIME0205
TIME0206
TIME0207
TIME0208
TIME0209
TIME0210
TIME0211
TIME0212
TIME0213
TIME0214
TIME0215
TIME0216
TIME0217
TIME0218
TIME0219
TIME0220
TIME0221
TIME0222
TIME0223
TIME0224
TIME0225
TIME0226
TIME0227
TIME0228
TIME0229
TIME0230

```

T I M E 0 2 3 1
 T I M E 0 2 3 2
 T I M E 0 2 3 3
 T I M E 0 2 3 4
 T I M E 0 2 3 5
 T I M E 0 2 3 6
 T I M E 0 2 3 7
 T I M E 0 2 3 8
 T I M E 0 2 3 9
 T I M E 0 2 4 0
 T I M E 0 2 4 1
 T I M E 0 2 4 2
 T I M E 0 2 4 3
 T I M E 0 2 4 4
 T I M E 0 2 4 5
 T I M E 0 2 4 6
 T I M E 0 2 4 7
 T I M E 0 2 4 8
 T I M E 0 2 4 9
 T I M E 0 2 5 0
 T I M E 0 2 5 1
 T I M E 0 2 5 2
 T I M E 0 2 5 3
 T I M E 0 2 5 4
 T I M E 0 2 5 5
 T I M E 0 2 5 6
 T I M E 0 2 5 7
 T I M E 0 2 5 8
 T I M E 0 2 5 9
 T I M E 0 2 6 0
 T I M E 0 2 6 1
 T I M E 0 2 6 2
 T I M E 0 2 6 3
 T I M E 0 2 6 4
 T I M E 0 2 6 5
 T I M E 0 2 6 6
 T I M E 0 2 6 7
 T I M E 0 2 6 8
 T I M E 0 2 6 9
 T I M E 0 2 7 0
 T I M E 0 2 7 1
 T I M E 0 2 7 2
 T I M E 0 2 7 3
 T I M E 0 2 7 4
 T I M E 0 2 7 5
 T I M E 0 2 7 6
 T I M E 0 2 7 7
 T I M E 0 2 7 8

```

NNHC=NUMBER OF NODES WHERE HEAT TRANSFER COEFFICIENT IS SPECIFIED
      NNHC=0
NTOTQ=TCTAL NUMBER OF KNOWN QX,QY,QZC, AND QZ
      NTOTQ=2*NNQXY+NNQZC+NNQZ
NTOTVP=TOTAL NUMBER OF KNOWN VELOCITIES, PRESSURES, AND TEMPERATURES
      NTOTVP=2*NNVELS+NNPS+NNNTS

PRINT ALL INPUT DATA
      WRITE(NWRITE,1035)NN,NE,NNCN
      WRITE(NWRITE,1036)NNVELS
      WRITE(NWRITE,1037)NNQXY
      WRITE(NWRITE,1038)NNPS
      WRITE(NWRITE,1039)NNNTS
      WRITE(NWRITE,1034)NNQZC
      WRITE(NWRITE,1040)NNQZ
      WRITE(NWRITE,1041)
      DO 150 I=1,NNCN
        WRITE(NWRITE,1045)NCN(I),XC(NCN(I)),YC(NCN(I))
      150 CONTINUE
      WRITE(NWRITE,1050)
      DO 155 I=1,NE
        WRITE(NWRITE,1055)I,NODE(I,1),NODE(I,2),NODE(I,3),
          1 NODE(I,4),NODE(I,5),NODE(I,6)
      155 CONTINUE
      WRITE(NWRITE,1060)
      DO 160 I=1,NNVELS
        WRITE(NWRITE,1065)I,NVS(I),X(NVS(I)),X(NVS(I))+NN)
      160 CONTINUE
      WRITE(NWRITE,1070)
      DO 165 I=1,NNQXY
        WRITE(NWRITE,1065)I,NQS(I),Q(NQS(I)),Q(NQS(I))+NN)
      165 CONTINUE
      WRITE(NWRITE,1080)
      DO 170 I=1,NNPS
        WRITE(NWRITE,1085)I,NPS(I),X(NCP(NPS(I)))
      170 CONTINUE
      WRITE(NWRITE,1081)
      DO 171 I=1,NNNTS
        WRITE(NWRITE,1085)I,NVS(I+NNVELS),X(NVS(I+NNVELS))+MM)
      171 CONTINUE
      WRITE(NWRITE,1083)

```

```

DO 173 I=1, NNQZC
WRITE(NWRITE,1085) I, NQS(I+NNQXY), Q(NQS(I+NNQXY)+2*NN)
CONTINUE
173 WRITE(NWRITE,1082)
DO 172 I=1, NNQZ
WRITE(NWRITE,1085) I, NQS(I+NNQXY+NNQZC), Q(NQS(I+NNQXY+NNQZC)+MM)
CONTINUE
172 T=0.00
DO CONTINUE
177 CONTINUE
DO 178 I=1, MMM
RHS(I)=0.00
DO 178 J=1, MMM
TM(I,J)=0.00
CONTINUE
178 IF(T.GT.0.00) GO TO 180
DO 179 I=1, MMM
CC(I,J)=0.00
CONTINUE
179 CONTINUE
DO 181 I=1, 21
DO 181 J=1, 21
TM$(I,J)=0.00
CC$(I,J)=0.00
CONTINUE
181

```

END OF INPUT AND VERIFICATION ROUTINE

```

DO 300 K=1, NE
N1=NODE(K,1)
N2=NODE(K,2)
N3=NODE(K,3)
N4=NODE(K,4)
N5=NODE(K,5)
N6=NODE(K,6)
N7=NODE(K,1)+NN
N8=NODE(K,2)+NN
N9=NODE(K,3)+NN
N10=NODE(K,4)+NN
N11=NODE(K,5)+NN
N12=NODE(K,6)+NN
N13=NCP(NODE(K,1))
N14=NCP(NODE(K,3))
N15=NCP(NODE(K,5))
N16=NODE(K,1)+MM
N17=NODE(K,2)+MM
N18=NODE(K,3)+MM
N19=NODE(K,4)+MM

```

TIME0279
TIME0280
TIME0281
TIME0282
TIME0283
TIME0284
TIME0285
TIME0286
TIME0287
TIME0288
TIME0289
TIME0290
TIME0291
TIME0292
TIME0293
TIME0294
TIME0295
TIME0296
TIME0297
TIME0298
TIME0299
TIME0300
TIME0301
TIME0302
TIME0303
TIME0304
TIME0305
TIME0306
TIME0307
TIME0308
TIME0309
TIME0310
TIME0311
TIME0312
TIME0313
TIME0314
TIME0315
TIME0316
TIME0317
TIME0318
TIME0319
TIME0320
TIME0321
TIME0322
TIME0323
TIME0324
TIME0325
TIME0326

```

N20=NODE(K,5)+MM
N21=NODE(K,6)+MM
XC$(1)=XC(NODE(K,1))
XC$(2)=XC(NODE(K,3))
XC$(3)=XC(NODE(K,5))
XC$(1)=XC(NODE(K,1))
XC$(2)=YC(NODE(K,3))
XC$(3)=YC(NODE(K,5))
A=1.DO
ZBAR=(YC$(1)+YC$(2)+YC$(3))/3.DO
RBAR=(XC$(1)+XC$(2)+XC$(3))/3.DO
IF(NCASE.EQ.2)A=RBAR
AA=1.DO
IF(NCASE.EQ.2)AA=2.DO*3.14159D0*(RBAR)
A1=XC$(2)*YC$(3)-XC$(3)*YC$(2)
A2=XC$(3)*YC$(1)-XC$(1)*YC$(3)
A3=XC$(1)*YC$(2)-XC$(2)*YC$(1)
B1=YC$(3)-YC$(1)
B2=YC$(1)-YC$(2)
C1=XC$(3)-XC$(2)
C2=XC$(1)-XC$(3)
C3=XC$(2)-XC$(1)
DEL=DABS(0.5D0*(XC$(1)+XC$(2)-YC$(3))+XC$(2)*(YC$(3)-YC$(1))
+XC$(3)*(YC$(1)-YC$(2)))
CONST=10.90D0*(1.DO*A/(3.DO*DEL))*AA
D1=-B1/6.DO
D2=-B2/6.DO
D3=-B3/6.DO
E1=-C1/6.DO
E2=-C2/6.DO
E3=-C3/6.DO
F1=B1/2520.DO
F2=B2/2520.DO
F3=B3/2520.DO
G1=C1/2520.DO
G2=C2/2520.DO
G3=C3/2520.DO
U1=T1(N1)
U2=T1(N2)
U3=T1(N3)
U4=T1(N4)
U5=T1(N5)
U6=T1(N6)
V1=T1(N7)
V2=T1(N8)
V3=T1(N9)
V4=T1(N10)

```

```

V5=TL(N11)
V6=TL(N12)
TM$(1,1)=0.75D0*(B1*B1+C1*C1)*CONST
TM$(1,2)=(B1*B2+C1*C2)*CONST
TM$(1,3)=-TM$(1,2)*0.25D0
TM$(1,4)=0
TM$(1,6)=(B1*B3+C1*C3)*CONST
TM$(1,5)=-TM$(1,6)*0.25D0
TM$(3,3)=.75D0*(B2*B2+C2*C2)*CONST
TM$(3,4)=(B2*B3+C2*C3)*CONST
TM$(3,5)=-TM$(3,4)*0.25D0
TM$(2,1)=TM$(1,2)
TM$(2,2)=TM$(1,2)
TM$(2,3)=8.D0/3.D0*(TM$(1,1)+TM$(3,3))+2.D0*TM$(1,2)
TM$(2,4)=2.D0*TM$(1,6)+TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(3,3)
TM$(2,5)=0
TM$(2,6)=TM$(1,6)+2.D0*TM$(3,4)+TM$(1,2)+4.D0/3.D0*TM$(1,1)
TM$(3,1)=TM$(1,3)
TM$(3,2)=TM$(1,3)
TM$(3,6)=0
TM$(5,5)=.75D0*(B3*B3+C3*C3)*CONST
TM$(4,1)=TM$(1,4)
TM$(4,2)=TM$(2,4)
TM$(4,3)=TM$(3,4)
TM$(4,4)=8.D0/3.D0*(TM$(3,3)+TM$(5,5))+2.D0*TM$(3,4)
TM$(4,5)=TM$(3,4)
TM$(4,6)=TM$(1,6)+TM$(3,4)+2.D0*TM$(1,2)+4.D0/3.D0*TM$(5,5)
TM$(6,3)=TM$(3,6)
TM$(5,1)=TM$(1,5)
TM$(5,2)=TM$(2,5)
TM$(5,3)=TM$(3,5)
TM$(5,4)=TM$(4,5)
TM$(5,6)=TM$(1,6)
TM$(6,1)=TM$(1,6)
TM$(6,2)=TM$(2,6)
TM$(6,4)=TM$(4,6)
TM$(6,5)=TM$(5,6)
TM$(6,6)=8.D0/3.D0*(TM$(5,5)+TM$(1,1))+2.D0*TM$(1,6)
IF(NCASE.NE.1) GO TO 3000

BEGIN INPUT OF NON-LINEAR TERMS

TM$(1,1)=TM$(1,1)
1-TM$(1,2)=TM$(1,1)+48.D0*U2-9.D0*U3+12.D0*U4-9.D0*U5+48.D0*U6)*F1
2- (78.D0*U1+48.D0*V2-9.D0*V3+12.D0*V4-9.D0*V5+48.D0*V6)*F1
3*G1
TM$(2,1)=TM$(2,1)
1-(48.D0*U1+160.D0*U2-32.D0*U3+16.D0*U4-20.D0*U5+80.D0*U6)*F1

```

```

TIME0375
TIME0376
TIME0377
TIME0378
TIME0379
TIME0380
TIME0381
TIME0382
TIME0383
TIME0384
TIME0385
TIME0386
TIME0387
TIME0388
TIME0389
TIME0390
TIME0391
TIME0392
TIME0393
TIME0394
TIME0395
TIME0396
TIME0397
TIME0398
TIME0399
TIME0400
TIME0401
TIME0402
TIME0403
TIME0404
TIME0405
TIME0406
TIME0407
TIME0408
TIME0409
TIME0410
TIME0411
TIME0412
TIME0413
TIME0414
TIME0415
TIME0416
TIME0417
TIME0418
TIME0419
TIME0420
TIME0421
TIME0422

```


TM\$(6,2)=TM\$(6,2)
 1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1
 2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.D0*V6)*G1
 3D0*U5+384.D0*U6)*F2
 4D0*U5+384.D0*U6)*F2
 5128.D0*V4-32.D0*V5+384.D0*V6)*G2
 TM\$(1,3)=TM\$(1,3)
 1-(-18.D0*U1-32.D0*U2-9.D0*U3-20.D0*U4+11.D0*U5-16.D0*U6)*F2
 2-(-18.D0*V1-32.D0*V2-9.D0*V3-20.D0*V4+11.D0*V5-16.D0*V6)*G2
 36)*G2
 TM\$(2,3)=TM\$(2,3)
 1-(-32.D0*U1+160.D0*U2+48.D0*U3+80.D0*U4-20.D0*U5+16.D0*U6)*F2
 2-(-32.D0*V1+160.D0*V2+48.D0*V3+80.D0*V4-20.D0*V5+16.D0*V6)*G2
 3TM\$(3,3)=TM\$(3,3)
 1-(-9.D0*U1+48.D0*U2+78.D0*U3+48.D0*U4-9.D0*U5+12.D0*U6)*F2
 2-(-9.D0*V1+48.D0*V2+78.D0*V3+48.D0*V4-9.D0*V5+12.D0*V6)*G2
 3TM\$(4,3)=TM\$(4,3)
 1-(-20.D0*U1+80.D0*U2+48.D0*U3+160.D0*U4-32.D0*U5+16.D0*U6)*F2
 2-(-20.D0*V1+80.D0*V2+48.D0*V3+160.D0*V4-32.D0*V5+16.D0*V6)*G2
 3TM\$(5,3)=TM\$(5,3)
 1-(-11.D0*U1-20.D0*U2-9.D0*U3-32.D0*U4-18.D0*U5-16.D0*U6)*F2
 2-(-11.D0*V1-20.D0*V2-9.D0*V3-32.D0*V4-18.D0*V5-16.D0*V6)*G2
 3TM\$(6,3)=TM\$(6,3)
 1-(-16.D0*U1+16.D0*U2+12.D0*U3+16.D0*U4-16.D0*U5-96.D0*U6)*F2
 2-(-16.D0*V1+16.D0*V2+12.D0*V3+16.D0*V4-16.D0*V5-96.D0*V6)*G2
 3TM\$(1,4)=TM\$(1,4)
 1-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F2
 2-(-24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6)*G2
 3TM\$(2,4)=TM\$(2,4)
 1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F2
 2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.D0*V6)*G2
 3D0*U5+128.D0*U6)*F3
 48.D0*U5+128.D0*U6)*F3
 5192.D0*V4-48.D0*V5+128.D0*V6)*G3
 TM\$(3,4)=TM\$(3,4)
 1-(-4.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4+16.D0*U5-48.D0*U6)*F2
 2-(-4.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4+16.D0*V5-48.D0*V6)*G2
 3D0*U5-16.D0*U6)*F3
 4D0*U5-16.D0*U6)*F3
 5+48.D0*V4-16.D0*V5-16.D0*V6)*G3

TME04771
 TME04772
 TME04773
 TME04774
 TME04775
 TME04776
 TME04777
 TME04778
 TME04779
 TME04780
 TME04811
 TME04812
 TME04823
 TME04845
 TME04866
 TME04887
 TME04888
 TME04889
 TME04900
 TME04911
 TME04923
 TME04934
 TME04945
 TME04956
 TME04967
 TME04978
 TME04989
 TME05000
 TME05011
 TME05023
 TME05034
 TME05045
 TME05056
 TME05067
 TME05078
 TME05089
 TME05100
 TME05112
 TME05123
 TME05134
 TME05145
 TME05156
 TME05167
 TME05178

TM\$(4,4)=TM\$(4,4)
 1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F2
 2-(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.D0*V6)*G2
 3D0*U5+128.D0*U6)*F3
 42.D0*U4-32.D0*V5+128.D0*V6)*G3
 5+384.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2
 TM\$(5,4)=TM\$(5,4)
 1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2
 2-(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0*V6)*G2
 3*U5-16.D0*U6)*F3
 4*U4+24.D0*V5-16.D0*V6)*G3
 5.D0*V4+24.D0*V5-16.D0*V6)*G3
 TM\$(6,4)=TM\$(6,4)
 1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F2
 2-(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.D0*V6)*G2
 3D0*U5+128.D0*U6)*F3
 46.D0*U4-16.D0*V5+128.D0*V6)*G3
 5+128.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3
 TM\$(1,5)=TM\$(1,5)
 1-(-18.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3
 2-(-18.D0*V1-16.D0*V2+11.D0*V3-20.D0*V4-9.D0*V5-32.D0*V6)*G3
 3*U5-16.D0*U6)*F3
 4*U4+24.D0*V5-16.D0*V6)*G3
 5.D0*V4+24.D0*V5-16.D0*V6)*G3
 TM\$(2,5)=TM\$(2,5)
 1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F2
 2-(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0*V6)*G2
 3D0*U5+128.D0*U6)*F3
 46.D0*U4-16.D0*V5+128.D0*V6)*G3
 5+128.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3
 TM\$(3,5)=TM\$(3,5)
 1-(-11.D0*U1-16.D0*U2-18.D0*U3-32.D0*U4-9.D0*U5-20.D0*U6)*F3
 2-(-11.D0*V1-16.D0*V2-18.D0*V3-32.D0*V4-9.D0*V5-20.D0*V6)*G3
 3*U5-16.D0*U6)*F3
 4*U4+24.D0*V5-16.D0*V6)*G3
 5.D0*V4+24.D0*V5-16.D0*V6)*G3
 TM\$(4,5)=TM\$(4,5)
 1-(-20.D0*U1+16.D0*U2-32.D0*U3+160.D0*U4+48.D0*U5+80.D0*U6)*F3
 2-(-20.D0*V1+16.D0*V2-32.D0*V3+160.D0*V4+48.D0*V5+80.D0*V6)*G3
 3D0*U5+128.D0*U6)*F3
 46.D0*U4-16.D0*V5+128.D0*V6)*G3
 5+128.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3
 TM\$(5,5)=TM\$(5,5)
 1-(-9.D0*U1+12.D0*U2-9.D0*U3+48.D0*U4+78.D0*U5+48.D0*U6)*F3
 2-(-9.D0*V1+12.D0*V2-9.D0*V3+48.D0*V4+78.D0*V5+48.D0*V6)*G3
 3*U5-16.D0*U6)*F3
 4*U4+24.D0*V5-16.D0*V6)*G3
 5.D0*V4+24.D0*V5-16.D0*V6)*G3
 TM\$(6,5)=TM\$(6,5)
 1-(-32.D0*U1+16.D0*U2-20.D0*U3+80.D0*U4+48.D0*U5+160.D0*U6)*F3
 2-(-32.D0*V1+16.D0*V2-20.D0*V3+80.D0*V4+48.D0*V5+160.D0*V6)*G3
 3D0*U5+128.D0*U6)*F3
 46.D0*U4-16.D0*V5+128.D0*V6)*G3
 5+128.D0*U1-16.D0*U2+11.D0*U3-20.D0*U4-9.D0*U5-32.D0*U6)*F3
 TM\$(1,6)=TM\$(1,6)
 1-(-24.D0*U1-16.D0*U2+4.D0*U3-48.D0*U4-16.D0*U5-32.D0*U6)*F1
 2-(-24.D0*V1-16.D0*V2+4.D0*V3-48.D0*V4-16.D0*V5-32.D0*V6)*G1
 3*U5+48.D0*U6)*F3
 4D0*U5+48.D0*U6)*F3
 516.D0*V4-16.D0*V5+48.D0*V6)*G3

```

TM$(2,6)=TM$(2,6)
1-(-16.D0*U1+128.D0*U2-16.D0*U3+128.D0*U4-16.D0*U5+128.D0*U6)*F1
2-(-16.D0*V1+128.D0*V2-16.D0*V3+128.D0*V4-16.D0*V5+128.D0*V6)*G1
3D0*U5+192.D0*U6)*F3
4D0*V4-48.D0*V5+192.D0*V6)*G3
5128.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4-16.D0*U5+128.D0*U6)*F3
TM$(3,6)=TM$(3,6)
1-(-16.D0*U1-16.D0*U2+24.D0*U3-32.D0*U4-16.D0*U5+128.D0*U6)*F3
2-(-16.D0*V1-16.D0*V2+24.D0*V3-32.D0*V4-48.D0*V5+128.D0*V6)*G3
3D0*U5-48.D0*U6)*F3
4D0*V4+4.D0*V5-48.D0*V6)*G3
516.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1
TM$(4,6)=TM$(4,6)
1-(-48.D0*U1+128.D0*U2-32.D0*U3+384.D0*U4+48.D0*U5+192.D0*U6)*F1
2-(-48.D0*V1+128.D0*V2-32.D0*V3+384.D0*V4+48.D0*V5+192.D0*V6)*G1
3D0*U5+128.D0*U6)*F3
46.D0*U4-16.D0*U5+128.D0*V5+128.D0*V6)*G3
5+128.D0*V4-16.D0*V5+128.D0*V6)*G3
TM$(5,6)=TM$(5,6)
1-(-16.D0*U1-16.D0*U2-16.D0*U3+48.D0*U4+120.D0*U5+48.D0*U6)*F1
2-(-16.D0*V1-16.D0*V2-16.D0*V3+48.D0*V4+120.D0*V5+48.D0*V6)*G1
3D0*U5-32.D0*U6)*F3
40*U5-32.D0*V5-32.D0*V6)*G3
56.D0*U4+24.D0*U5+128.D0*U6)*F3
TM$(6,6)=TM$(6,6)
1-(-32.D0*U1+128.D0*U2-48.D0*U3+192.D0*U4+48.D0*U5+384.D0*U6)*F1
2-(-32.D0*V1+128.D0*V2-48.D0*V3+192.D0*V4+48.D0*V5+384.D0*V6)*G1
3D0*U5+384.D0*U6)*F3
4D0*U5+384.D0*V6)*G3
5128.D0*U4-32.D0*U5+384.D0*V6)*G3

THIS ENDS ADDITION OF NON-LINEAR TERMS TO THE LOCAL ARRAY

3000 CONTINUE
TM$(7,7)=TM$(1,1)
TM$(7,8)=TM$(1,2)
TM$(7,9)=TM$(1,3)
TM$(7,10)=TM$(1,4)
TM$(7,11)=TM$(1,5)
TM$(7,12)=TM$(1,6)
TM$(8,7)=TM$(2,1)
TM$(8,8)=TM$(2,2)
TM$(8,9)=TM$(2,3)
TM$(8,10)=TM$(2,4)
TM$(8,11)=TM$(2,5)
TM$(8,12)=TM$(2,6)
TM$(9,7)=TM$(3,1)
TM$(9,8)=TM$(3,2)

```


| | | | |
|------------|----|------|--------|
| TM\$(1,16) | =1 | .DO* | CONST2 |
| TM\$(2,17) | =1 | .DO* | CONST2 |
| TM\$(3,18) | =1 | .DO* | CONST2 |
| TM\$(4,19) | =1 | .DO* | CONST2 |
| TM\$(5,20) | =1 | .DO* | CONST2 |
| TM\$(6,21) | =1 | .DO* | CONST2 |

ALPHA1=ALPHA/VISCOSITY

ALPHA1=9.4454D-05

ALPHA1=9.4454D-05
ALPHA1=ALPHA1
CONST1=16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,63,64,65,66,67,68,69,70,71,72,73,74,75,76,77,78,79,80,81,82,83,84,85,86,87,88,89,90,91,92,93,94,95,96,97,98,99,100,101,102,103,104,105,106,107,108,109,110,111,112,113,114,115,116,117,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,140,141,142,143,144,145,146,147,148,149,150,151,152,153,154,155,156,157,158,159,160,161,162,163,164,165,166,167,168,169,170,171,172,173,174,175,176,177,178,179,180,181,182,183,184,185,186,187,188,189,190,191,192,193,194,195,196,197,198,199,200,201,202,203,204,205,206,207,208,209,210,211,212,213,214,215,216,217,218,219,220,221,222,223,224,225,226,227,228,229,230,231,232,233,234,235,236,237,238,239,240,241,242,243,244,245,246,247,248,249,250,251,252,253,254,255,256,257,258,259,260,261,262,263,264,265,266,267,268,269,270,271,272,273,274,275,276,277,278,279,280,281,282,283,284,285,286,287,288,289,290,291,292,293,294,295,296,297,298,299,300,301,302,303,304,305,306,307,308,309,310,311,312,313,314,315,316,317,318,319,320,321,322,323,324,325,326,327,328,329,330,331,332,333,334,335,336,337,338,339,340,341,342,343,344,345,346,347,348,349,350,351,352,353,354,355,356,357,358,359,360,361,362,363,364,365,366,367,368,369,370,371,372,373,374,375,376,377,378,379,380,381,382,383,384,385,386,387,388,389,390,391,392,393,394,395,396,397,398,399,400,401,402,403,404,405,406,407,408,409,410,411,412,413,414,415,416,417,418,419,420,421,422,423,424,425,426,427,428,429,430,431,432,433,434,435,436,437,438,439,440,441,442,443,444,445,446,447,448,449,450,451,452,453,454,455,456,457,458,459,460,461,462,463,464,465,466,467,468,469,470,471,472,473,474,475,476,477,478,479,480,481,482,483,484,485,486,487,488,489,490,491,492,493,494,495,496,497,498,499,500,501,502,503,504,505,506,507,508,509,510,511,512,513,514,515,516,517,518,519,520,521,522,523,524,525,526,527,528,529,530,531,532,533,534,535,536,537,538,539,540,541,542,543,544,545,546,547,548,549,550,551,552,553,554,555,556,557,558,559,560,561,562,563,564,565,566,567,568,569,570,571,572,573,574,575,576,577,578,579,580,581,582,583,584,585,586,587,588,589,590,591,592,593,594,595,596,597,598,599,600,601,602,603,604,605,606,607,608,609,610,611,612,613,614,615,616,617,618,619,620,621,622,623,624,625,626,627,628,629,630,631,632,633,634,635,636,637,638,639,640,641,642,643,644,645,646,647,648,649,650,651,652,653,654,655,656,657,658,659,660,661,662,663,664,665,666,667,668,669,670,671,672,673,674,675,676,677,678,679,680,681,682,683,684,685,686,687,688,689,690,691,692,693,694,695,696,697,698,699,700,701,702,703,704,705,706,707,708,709,710,711,712,713,714,715,716,717,718,719,720,721,722,723,724,725,726,727,728,729,730,731,732,733,734,735,736,737,738,739,740,741,742,743,744,745,746,747,748,749,750,751,752,753,754,755,756,757,758,759,760,761,762,763,764,765,766,767,768,769,770,771,772,773,774,775,776,777,778,779,780,781,782,783,784,785,786,787,788,789,790,791,792,793,794,795,796,797,798,799,800,801,802,803,804,805,806,807,808,809,810,811,812,813,814,815,816,817,818,819,820,821,822,823,824,825,826,827,828,829,830,831,832,833,834,835,836,837,838,839,840,841,842,843,844,845,846,847,848,849,850,851,852,853,854,855,856,857,858,859,860,861,862,863,864,865,866,867,868,869,870,871,872,873,874,875,876,877,878,879,880,881,882,883,884,885,886,887,888,889,890,891,892,893,894,895,896,897,898,899,900,901,902,903,904,905,906,907,908,909,910,911,912,913,914,915,916,917,918,919,920,921,922,923,924,925,926,927,928,929,930,931,932,933,934,935,936,937,938,939,940,941,942,943,944,945,946,947,948,949,950,951,952,953,954,955,956,957,958,959,960,961,962,963,964,965,966,967,968,969,970,971,972,973,974,975,976,977,978,979,980,981,982,983,984,985,986,987,988,989,990,991,992,993,994,995,996,997,998,999,1000,1001,1002,1003,1004,1005,1006,1007,1008,1009,1010,1011,1012,1013,1014,1015,1016,1017,1018,1019,1020,1021,1022,1023,1024,1025,1026,1027,1028,1029,1030,1031,1032,1033,1034,1035,1036,1037,1038,1039,1040,1041,1042

```

CONST3=-DEL/180.D0
CD$(1,1)=6.D0*CONST3
CD$(1,2)=0.D0
CD$(1,3)=-CONST3
CD$(1,4)=-4.D0*CONST3
CD$(1,5)=-CONST3
CD$(1,6)=0.D0
CD$(2,1)=0.D0
CD$(2,2)=32.D0*CONST3
CD$(2,3)=0.D0
CD$(2,4)=16.D0*CONST3
CD$(2,5)=16.D0*CONST3
CD$(2,6)=16.D0*CONST3
CD$(3,1)=0.D0
CD$(3,2)=0.D0*CONST3
CD$(3,3)=0.D0
CD$(3,4)=0.D0
CD$(3,5)=-CONST3
CD$(3,6)=-4.D0*CONST3
CD$(4,1)=16.D0*CONST3
CD$(4,2)=0.D0
CD$(4,3)=32.D0*CONST3
CD$(4,4)=0.D0
CD$(4,5)=16.D0*CONST3
CD$(4,6)=-CONST3
CD$(5,1)=-4.D0*CONST3
CD$(5,2)=-CONST3
CD$(5,3)=0.D0
CD$(5,4)=6.D0*CONST3
CD$(5,5)=0.D0
CD$(5,6)=0.D0
CD$(6,1)=0.D0
CD$(6,2)=16.D0*CONST3
CD$(6,3)=16.D0*CONST3
CD$(6,4)=16.D0
CD$(6,5)=32.D0*CONST3
CD$(6,6)=CD$(1,1)
CD$(7,7)=CD$(1,2)
CD$(7,8)=CD$(1,3)
CD$(7,9)=CD$(1,4)
CD$(7,10)=CD$(1,5)
CD$(7,11)=CD$(1,6)
CD$(7,12)=CD$(2,1)
CD$(7,13)=CD$(2,2)
CD$(7,14)=CD$(2,3)
CD$(7,15)=CD$(2,4)
CD$(7,16)=CD$(2,5)
CD$(8,8)=CD$(8,9)
CD$(8,10)=CD$(8,11)
CD$(8,12)=CD$(8,13)
CD$(8,14)=CD$(8,15)
CD$(8,16)=CD$(8,17)

```

```

TIME0759
TIME0760
TIME0761
TIME0762
TIME0763
TIME0764
TIME0765
TIME0766
TIME0767
TIME0768
TIME0769
TIME0770
TIME0771
TIME0772
TIME0773
TIME0774
TIME0775
TIME0776
TIME0777
TIME0778
TIME0779
TIME0780
TIME0781
TIME0782
TIME0783
TIME0784
TIME0785
TIME0786
TIME0787
TIME0788
TIME0789
TIME0790
TIME0791
TIME0792
TIME0793
TIME0794
TIME0795
TIME0796
TIME0797
TIME0798
TIME0799
TIME0800
TIME0801
TIME0802
TIME0803
TIME0804
TIME0805
TIME0806

```


TIME0903
TIME0904
TIME0905
TIME0906
TIME0907
TIME0908
TIME0909
TIME0910
TIME0911
TIME0912
TIME0913
TIME0914
TIME0915
TIME0916
TIME0917
TIME0918
TIME0919
TIME0920
TIME0921
TIME0922
TIME0923
TIME0924
TIME0925
TIME0926
TIME0927
TIME0928
TIME0929
TIME0930
TIME0931
TIME0932
TIME0933
TIME0934
TIME0935
TIME0936
TIME0937
TIME0938
TIME0939
TIME0940
TIME0941
TIME0942
TIME0943
TIME0944
TIME0945
TIME0946
TIME0947
TIME0948
TIME0949
TIME0950

```

300 CONTINUE
    IF(NNQXY.EQ.0) GO TO 310
    DO 310 I=1, NNQXY
        RHS(NQS(I))=RHS(NQS(I))+Q(NQS(I))+65.962D0
    RHS(NQS(I))+NN)=RHS(NQS(I)+NN)+Q(NQS(I)+NN)
    CONTINUE
310 IF(NNQZC.EQ.0) GO TO 312
    DO 312 I=1, NNQZC
        RHS(NQS(NNQXY+I)+2*NN)=RHS(NQS(NNQXY+I)+2*NN)+Q(NQS(NNQXY+I)+2*NN)
    CONTINUE
312 IF(NNQZ.EQ.0) GO TO 311
    DO 311 I=1, NNQZ
        RHS(NQS(NNQXY+NNQZC+I)+MM)=RHS(NQS(NNQXY+NNQZC+I)+MM)+
        1Q(NQS(NNQXY+NNQZC+I)+MM)
    CONTINUE
311

```

MODIFICATION OF RHS FOR TM AND CD BOUNDARY CONDITIONS

```

DO 315 I=1, MMM
DO 315 J=1, NTOTVP
    JX=NVIS(J)
    RHS(I)=RHS(I)-TM(I, JX)*X(JX)
    TM(I, JX)=0.D0
    TM(JX, I)=0.D0
CONTINUE
315 IF(T.GT.0.D0) GO TO 321
    DO 320 I=1, NTOTVP
        K=NVIS(I)
        Y(1, K)=X(K)
    DO 316 J=1, MMM
        CD(J, K)=0.D0
        CD(K, J)=0.D0
        CD(K, K)=1.D0
        RHS(K)=0.D0
    CONTINUE
316 NY=117
    NL=0
    M=70
    JSKF=0
    MAXDER=6
    IPRT=1
    HM=1.D-08
    HMIN=1.D-12
    HMAX=5.D-02
    RMSEPS=1.D-03
    IF(T-0.D0) 323, 323, 321
323 GO TO 325

```

```

321 TEND=T
325 CONTINUE
      CALL SDESOL(Y,YL,T,TEND,NY,NL,M,JSKF,MAXDER,IPRT,H,HMIN,HMAX,
      1 RMSEPS,W)
      IF(T.GT.5.D-02) GO TO 324
      DO 322 J=1,MMM
      TDIFF=DABS(T1(J)-Y(1,J))
      T1(J)=Y(1,J)
      EPSLN=1.D-06
      IF(TDIFF-EPSLN) 322,177,177
322 CONTINUE
324 WRITE(NWRITE,2000)
      DO 360 I=1,MMM
      WRITE(NWRITE,2005) I,Y(1,I)
360 CONTINUE
      WRITE(NWRITE,2021)
      WRITE(NWRITE,2025)
      WRITE(NWRITE,2030)
      WRITE(NWRITE,2035)
      WRITE(NWRITE,2040)
      WRITE(NWRITE,2045)
      WRITE(NWRITE,2050)
      WRITE(NWRITE,2055)
      FORMAT(110)
500 FCRMAT(1H,18X,'TIME-DEPENDENT FLUID MECHANICS PROBLEM',////)
600 FCRMAT(3I10)
1005 FCRMAT(6X,A4,I10,2F10.0)
1010 FCRMAT(7I10)
1015 FCRMAT(6X,A4,I10,F10.0)
1016 FCRMAT(6X,A4,I10)
1020 FCRMAT(110,2F10.0)
1025 FCRMAT(6X,A4,I10,2F10.0)
1030 FCRMAT(6X,A4,2I10,2F10.0)
1034 FCRMAT(5X,'NNQZC=',I3,/)
1035 FCRMAT(5X,'NO. OF CORNERS=',I3,/)
1036 FCRMAT(5X,'NNVELS=',I3,/)
1037 FCRMAT(5X,'NNQXY=',I3,/)
1038 FCRMAT(5X,'NNPS=',I3,/)
1039 FCRMAT(5X,'NNTS=',I3,/)
1040 FCRMAT(5X,'NNQZ=',I3,/)
1041 FCRMAT(5X,'SUMMARY OF NODAL COORDINATES',/,/,
      17X,I,12X,X(I),13X,Y(I),/)
1045 FCRMAT(5X,I3,13,2(7X,F10.3))
1050 FCRMAT(5X,'LISTING OF SYSTEM TOPOLOGY',/,/,5X
      1,ELEMENT NUMBER,20X,'NODE NUMBERS',/)
1055 FCRMAT(5X,I3,10X,6(5X,I3))

```

```

TIME0951
TIME0952
TIME0953
TIME0954
TIME0955
TIME0956
TIME0957
TIME0958
TIME0959
TIME0960
TIME0961
TIME0962
TIME0963
TIME0964
TIME0965
TIME0966
TIME0967
TIME0968
TIME0969
TIME0970
TIME0971
TIME0972
TIME0973
TIME0974
TIME0975
TIME0976
TIME0977
TIME0978
TIME0979
TIME0980
TIME0981
TIME0982
TIME0983
TIME0984
TIME0985
TIME0986
TIME0987
TIME0988
TIME0989
TIME0990
TIME0991
TIME0992
TIME0993
TIME0994
TIME0995
TIME0996
TIME0997

```



```

80 BLOCK OF AUXILLARY STORAGE, AND OBTAIN INITIAL VALUES OF
90 DERIVATIVES.
100 THE CALLING SEQUENCE FOR SDESOL IS
110
120 CALL SDESOL(Y,YL,T,TEND,N,NL,M,JSKF,MAXDER,IPRT,H,HMIN,HMAX,RMSEPS,W)
130
140 WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.
150
160 Y - ARRAY DIMENSIONED (7,NY). THIS ARRAY CONTAINS THE
170 DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
180 Y(J+1,I) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VARIABLE.
190 IABLE TIMES H*J/J-FACTORIAL, WHERE H IS THE CURRENT
200 STEPSIZE. ON FIRST ENTRY THE CALLER SUPPLIES THE
210 INITIAL VALUES OF EACH VARIABLE IN Y(1,I). ON SUB-
220 SEQUENT ENTRIES IT IS ASSUMED THE ARRAY HAS NOT
230 BEEN CHANGED. TO INTERPOLATE TO NON-MESH POINTS,
240 THESE VALUES CAN BE USED AS FOLLOWS. IF H IS THE
250 CURRENT STEPSIZE AND VALUES AT TIME T+H ARE
260 NEEDED, LET S = E/H AND THEN
270
280 I-TH VARIABLE AT T+H IS SUM Y(J+1,I)*S**J
290 J=0
300
310 THE VALUE OF JS IS OBTAINED IN THE CALLING PROGRAM
320 BY JS = IABS(JSKF/10), WHICH APPEAR LINEARLY.
330 ARRAY OF NL VARIABLES, WHICH APPEAR LINEARLY.
340 - CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME)
350 - END TIME OF DIFFERENTIAL EQUATIONS AND NONLINEAR
360 - VARIABLES.
370 - NUMBER OF LINEAR VARIABLES
380 - NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST
390 - AN INDICATOR USED BOTH ON INPUT AND OUTPUT
400 - ON INPUT, JSKF = -1 INDICATES A RESTART CALL TO
410 SDESOL. JSKF = 0 INDICATES AN INITIAL CALL TO THE
420 SDESOL. JSKF > 0 INDICATES A CONTINUATION OF THE
430 PREVIOUS CALL TO SDESOL. JSKF < -1 MAY HAVE RETURNS
440 FROM THE SOL. CALL TO SDESOL. JSKF < -1 MAY HAVE RETURNS
450 FROM THE SOL. CALL TO SDESOL. JSKF < -1 MAY HAVE RETURNS
460 RESULTS IN TERMINATION.
470 APPROPRIATE COMMENT.
480 ON OUTPUT, JSKF CONSISTS OF TWO DIGITS AND SIGN,
490 + OR - QP. Q IS THE ORDER OF THE FORMULA CURRENTLY
500 BEING USED. P INDICATES THE TYPE OF RETURN, AS
510 FOLLOWS.
520 JSKF > 0, P = 1 IS THE NORMAL RETURN
530 JSKF < 0, P IS AN ERROR RETURN, WITH THE FOLLOWING
540
550

```

```

MEANINGS.
P = 1      ERROR TEST FAILED FOR H > HMIN
P = 3      CORRECTOR FAILED TO CONVERGE FOR H > HMIN
P = 4      CORRECTOR FAILED TO CONVERGE FOR FIRST
          ORDER METHOD
P = 5      ERROR RETURN FROM SUBROUTINE NUTSL
P = 6      ERROR RETURN FROM SUBROUTINE Derval
          METHOD. IT MUST BE NO GREATER THAN SIX.
          INTERNAL PRINT NO PRINT CONTROL INDICATOR FOR LCASUB.
          IPRT = 0
          IPRT > 0
          PRINT COUNTERS, STEPSIZE, CURRENT TIMES
          AND VALUES OF DEPENDENT VARIABLES AT
          EACH STEP.
          CURRENT STEPSIZE. THE ONE INITIAL VALUE MUST BE SUPPLIED
          BUT NEED NOT BE THOSE WHICH MUST BE USED, SINCE THE
          SUBROUTINE WILL CHOOSE A SMALLER ONE IF NECESSARY TO
          KEEP THE ERROR PER STEP SMALLER THAN THE SPECIFIED
          VALUE. IT IS BETTER TO UNDERESTIMATE THE INITIAL
          STEPSIZE THAN TO OVERESTIMATE IT. THE STEPSIZE IS
          NORMALLY NOT CHANGED BY THE USER.
          MINIMUM STEPSIZE ALLOWED
          MAXIMUM STEPSIZE ALLOWED
          THE ERROR TEST ERROR TO CURRENT ESTIMATES, ER(I), DIVIDED BY
          THE SINGLE STEP ERROR TO CURRENT SIZE AND/OR THE ORDER
          YMAX(I) = (MAXIMUM THE STEPSIZE
          LESS THAN EPS. ACHIEVE THIS.
          ARE VARIED TO STORAGE ARRAY. MUST BE AT LEAST 13*NY + 5*NL
          SCRATCH STORAGE PLUS THOSE REQUIRED FOR STORAGE OF THE
          LOCATIONS, (SEE DESCRIPTION OF SUBROUTINE JACMAT).
          MATRIX PW (THE STORAGE OF PW WILL NORMALLY REQUIRE NO MORE THAN
          THE STORAGE OF PW LOCATIONS, AND IF COMPACT STORAGE TECH-
          Niques ARE USED, CAN BE MUCH FEWER.
          -----
IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION Y(7,1), YL(1), W(1)
IF (JSKF.GT.0) GO TO 120
IF (JSKF.LT.-1) GO TO 140
N = NY+NL
IF (JSKF.LT.0) GO TO 110
          IF THIS IS THE FIRST ENTRY, OBTAIN VALUES OF THE DERIVATIVES.
          CALL Derval (Y, YL, T, N, NY, W, KRETR)
          IF (KRETR.NE.0) GO TO 130
          NOW SET UP STORAGE BLOCKS IN THE W ARRAY. THIS NEEDS TO BE DONE

```

```

          SDE 970
          SDE 980
          SDE 1010
          SDE 1020

```

1030 SDE
1040 SDE
1050 SDE
1060 SDE
1070 SDE
1080 SDE
1090 SDE
1100 SDE
1110 SDE
1120 SDE
1130 SDE

ONLY INITIALLY AND ON RESTARTS.
THE ARRAY SAVE STARTS AT LOCATION
THE ARRAY YLSV STARTS AT LOCATION
THE ARRAY YMAX STARTS AT LOCATION
THE ARRAY ESV STARTS AT LOCATION
THE ARRAY FI STARTS AT LOCATION
THE ARRAY DY STARTS AT LOCATION
THE MATRIX PW STARTS AT LOCATION
1 NSVL = 7*NY+1
NYMAX = NSVL+NL
NER = NYMAX+NY
NESV = NER+NY
NFI = NESV+NY
NDY = NFI+N
NPW = NDY+N
JS = JSKF
CALL LDASUB (Y,YL,T,TEND,N,NY,M,JS,KE,MAXDER,IPRT,H,HMIN,HMAX,
1 RMSEPS,W,W(NSVL),W(NYMAX),W(NER),W(NESV),W(NFI),W(NDY),W(NPW))
CODE JSKF ON RETURN FROM LDASUB
JSKF = ISIGN(JS*10+IABS(KF),KF)
RETURN
130 JSKF = -6
RETURN
140 PRINT 1, JSKF
STOP

CCCCCCCCCCCC

1240 SDE
1250 SDE
1260 SDE

1 FORMAT ('OIT IS AN ERROR TO ENTER SDESOL WITH JSKF = ',I10//
1 RUN HAS BEEN TERMINATED.')
END
SUBROUTINE LDASUB (Y,YL,T,TEND,N,NY,M,JSTART,KFLAG,MAXOR,IPRT,H,
1 HMIN,HMAX,RMSEPS,SAVE,YLSV,YMAX,ER,ESV,FI,DY,PW)
SUBROUTINE LDASUB IS A MODIFICATION OF SUBROUTINE DFASUB
WHICH IS DUE TO R. L. BROWN AND C. W. GEAR. DFASUB IS DOCUMENTED
IN THE REPORT
DOCUMENTATION FOR DFASUB--
BY R. L. BROWN AND C. W. GEAR
REPORT UIUCDCS-R-73-575, JULY 1973
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN
URBANA, ILLINOIS 61801
THIS REPORT IS AVAILABLE FROM THE NATIONAL TECHNICAL INFORMATION
SERVICE OF THE U. S. DEPARTMENT OF COMMERCE UNDER ACCESSION NUMBER

CCCC

1330 SDE
1340 SDE

130 JSKF = -6
RETURN
140 PRINT 1, JSKF
STOP

CCCC

30 LDA
40 LDA
50 LDA
60 LDA
70 LDA
80 LDA
90 LDA
100 LDA
110 LDA
120 LDA
130 LDA

130 JSKF = -6
RETURN
140 PRINT 1, JSKF
STOP

CCCCCCCCCCCC

COO-1469-225.

THE MODIFICATION HERE IS DOCUMENTED IN THE REPORT
A PROGRAM FOR THE NUMERICAL SOLUTION OF LARGE SPARSE SYSTEMS OF
ALGEBRAIC AND IMPLICITLY DEFINED STIFF DIFFERENTIAL EQUATIONS
BY RICHARD FRANK
REPORT NPS53FE76051, MAY 1976
NAVAL POSTGRADUATE SCHOOL
MONTEREY, CALIFORNIA 93940

THE CALLING SEQUENCE FOR LDASUB IS

CALL LDASUB(Y, YL, T, TEND, N, NY, M, JSTART, KFLAG, MAXOR, IPRT, H, HMIN,
HMAX, RMSEPS, SAVE, YLSV, YMAX, ER, ESV, FL, DY, PW)

WHERE THE PARAMETERS ARE DEFINED AS FOLLOWS.
- ARRAY DIMENSIONED (7, NY). THIS ARRAY CONTAINS THE
DEPENDENT VARIABLES AND THEIR SCALED DERIVATIVES.
Y(J+1, I) CONTAINS THE J-TH DERIVATIVE OF THE I-TH VARIABLE
IABLE SIZE. H*J/J-FACTORIAL, WHERE H IS THE CURRENT
STEPSIZE. ON FIRST ENTRY THE CALLER SUPPLIES THE
INITIAL VALUES OF EACH VARIABLE OF THE DERIVATIVES
IN Y(2, I). ON SUBSEQUENT ENTRIES IT IS ASSUMED THAT
THE ARRAY Y HAS NOT BEEN CHANGED. TO INTERPOLATE TO
NON-MESH POINTS, THESE VALUES CAN BE USED AS FOLLOWS.
IF H IS THE CURRENT STEP SIZE AND VALUES AT TIME T+E
NEEDED, LET S = E/H AND THEN

I-TH VARIABLE AT T+E IS
SUM Y(J+1, I)*S**J
J=0

THE VALUE OF NQ IS OBTAINED IN THE CALLING PROGRAM
BY NQ = JSTART.

- YL - ARRAY OF NL = N - NY VARIABLES WHICH APPEAR LINEARLY.
- T - THE USER SUPPLIES INITIAL VALUES FOR THESE VARIABLES.
- TEND - CURRENT VALUE OF THE INDEPENDENT VARIABLE (TIME)
- N - END TIME
- NY - TOTAL NUMBER OF VARIABLES
- M - NUMBER OF DIFFERENTIAL EQUATIONS AND NONLINEAR
VARIABLES.
- NUMBER OF VARIABLES INCLUDED IN THE ERROR TEST.
 THIS NUMBER CAN BE NO GREATER THAN NY. IF IT IS
 GREATER THAN NY, NY VARIABLES ARE USED IN THE ERROR

LDA 140
LDA 150
LDA 160
LDA 170
LDA 180
LDA 190
LDA 200
LDA 210
LDA 220
LDA 230
LDA 240
LDA 250
LDA 260
LDA 270
LDA 280
LDA 290
LDA 300
LDA 310
LDA 320
LDA 330
LDA 340
LDA 350
LDA 360
LDA 370
LDA 380
LDA 390
LDA 400
LDA 410
LDA 420
LDA 430
LDA 440
LDA 450
LDA 460
LDA 470
LDA 480
LDA 490
LDA 500
LDA 510
LDA 520
LDA 530
LDA 540
LDA 550
LDA 560
LDA 570
LDA 580
LDA 590
LDA 600
LDA 610

```

JSTART - TEST* AND OUTPUT INDICATOR. FOLLOWING MEANINGS. PREVIOUS
          INPUT ON INPUT <0
          THIS INDICATES A TERMINATION FROM A PREVIOUS
          POINT FOLLOWING A TERMINATION OF THE RUN OR
          SOLUTION OF ANOTHER PROBLEM DURING THE SAME
          RUN. PARAMETERS IN THE CALLING SEQUENCE
          MUST HAVE BEEN PRESERVED FROM THE PREVIOUS
          USE, PARTICULARLY THE ARRAYS
          SAVE, YLSV, ES, V, AND PW. AFTER A CALL
          TO SUBROUTINE LDASAV, WHICH ALSO SAVES
          NECESSARY PARAMETERS, INTERNAL TO LDASUB. THE
          INDICATES AN INITIAL CALL TO LDASUB. THE
          ROUTINE INITIALIZES ITSELF, SCALES THE
          DERIVATIVES IN Y(2:I) AND THEN PERFORMS THE
          INTEGRATION UNTIL TEND. TO BE CONTINUED.
          INDICATES THE SOLUTION IS TO BE CONTINUED.
          AFTER THE INITIAL ENTRY IT IS NEITHER
          DESIRABLE NOR NECESSARY TO RE-INITIALIZE
          JSTART = 0, SINCE THIS RE-INITIALIZES
          THE CODE, BEGINNING WITH A FIRST ORDER
          METHOD AGAIN. SET TO THE VALUE OF NQ, THE
          ORDER OF THE FORMULA CURRENTLY BEING USED. THE
          ORDER OF THE FORMULA INDICATOR, WITH THE FOLLOWING
          MEANINGS
          +1 THE INTEGRATION WAS SUCCESSFUL
          -1 ERROR TEST FAILED FOR H > HMIN
          -3 CORRECTOR FAILED TO CONVERGE FOR H > HMIN
          -4 ORDER METHOD FAILED TO CONVERGE FOR FIRST
          ORDER METHOD
          -5 ERROR RETURN FROM SUBROUTINE NUTSL
          MAXOR - ORDER DERIVATIVE THAT SHOULD BE USED IN THE
          METHOD. IT MUST BE NO GREATER THAN SIX. IF IT IS
          GREATER THAN SIX, THE MAXIMUM ORDER USED WILL BE SIX.
          IPRT - INTERNAL PRINT CONTROL INDICATOR
          = 0 NO PRINT
          > 0 PRINT COEFFICIENTS, CURRENT TIME
          AND VALUES OF DEPENDENT VARIABLES AT
          EACH STEP. AN INITIAL VALUE MUST BE SUPPLIED
          CURRENT STEP SIZE. THE ONE WHICH WILL BE USED, SINCE THE
          BUT NEED NOT BE CHOSEN A SMALLER THAN THE SPECIFIED
          SUBROUTINE WILL PERFORM STEP TO UNDERESTIMATE THE INITIAL
          KEEP THE ERROR IT IS BETTER TO UNDERESTIMATE THE INITIAL
          VALUE. IT IS BETTER TO UNDERESTIMATE THE INITIAL
          STEP SIZE THAN TO OVERESTIMATE IT. THE STEP SIZE IS
          NORMALLY NOT CHANGED BY THE USER.

```

```

LDA 620
LDA 630
LDA 640
LDA 650
LDA 660
LDA 670
LDA 680
LDA 690
LDA 700
LDA 710
LDA 720
LDA 730
LDA 740
LDA 750
LDA 760
LDA 770
LDA 780
LDA 790
LDA 800
LDA 810
LDA 820
LDA 830
LDA 840
LDA 850
LDA 860
LDA 870
LDA 880
LDA 890
LDA 900
LDA 910
LDA 920
LDA 930
LDA 940
LDA 950
LDA 960
LDA 970
LDA 980
LDA 990
LDA 1000
LDA 1010
LDA 1020
LDA 1030
LDA 1040
LDA 1050
LDA 1060
LDA 1070
LDA 1080
LDA 1090

```

```

HMIN  STEPSIZE ALLOWED 1100
HMAX  STEPSIZE ALLOWED 1110
RMSEPS  TEST CONSTANT 1110
        THE SINGLE STEP ERROR ESTIMATES, ER(I), DIVIDED BY 1120
        YMAX(I) = (MAXEPS * CURRENT TIME OF Y(I)) MUST BE 1130
        LESS THAN RMSEPS. THIS. 1140
        AN ARRAY OF LENGTH AT LEAST 7*NY 1150
        AN ARRAY OF LENGTH AT LEAST NL 1160
        A VECTOR OF LENGTH NY ON THE FIRST CALL, THESE WILL 1170
        BE INITIALISED AS YMAX(I) = MAX(1, Y(1, I)), 1180
        A VECTOR OF LENGTH NY = NY + NL 1190
        A VECTOR OF LENGTH N = NY + NL 1200
        AN ARRAY IN WHICH THE J MATRIX COMPUTED 1210
        IN SUBROUTINE JACMAT WILL BE STORED. SIZE WHICH 1220
        MUST BE ALLOWED IS, BUT NORMALLY WON'T BE MORE THAN 1230
        N**2 + 2*N LOCATIONS. THE LATTER 2*N BEING REQUIRED 1240
        BY THE LINEAR EQUATION SOLVER. 1250
        1260
        1270
        1280
        1290
        1300
        1310
        1320
        1330

```

```

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION Y(7,1), YL(1), SAVE(7,1), YMAX(1), ER(1), YLSV(1), F1(1)
PERT(6,3), COF(21), ESV(1), PW(1), SAV(1), A(29)
EQUIVALENCE (A(8),BND), (A(9),BR), (A(10),E), (A(11),EDWN),
1(A(12),ENQ1), (A(13),ENQ2), (A(14),ENQ3), (A(15),EPS), (A(16),EUP)
2(A(17),HNEW), (A(18),PEPSH), (A(19),IDDOB), (A(20),IWEVAL),
3(A(21),K), (A(22),LCOPYL), (A(23),LCOPYR), (A(24),MAXDER),
4(A(25),M1), (A(26),NL), (A(27),NQ), (A(28),NS), (A(29),NW)

```

```

LDA 1410
LDA 1420
LDA 1430
LDA 1440
LDA 1450
LDA 1460
LDA 1470

```

```

THE COEFFICIENTS IN THE PERT ARRAY ARE USED FOR ERROR TESTING AND
CHANGING STEPSIZE AND NEED TO BE ACCURATE TO ONLY A FEW DIGITS.

DATA PERT/4.0,9.0,16.0,25.0,36.0,49.0,9.0,16.0,25.0,36.0,
1,49.0,64.0,1.0,1.0,1.0,2.5,2.7889D-2,1.70569D-3,6.83929D-5/

THE ENTRIES IN THE COF ARRAY ARE THE COEFFICIENTS FOR THE STIFFLY
STABLE METHODS USED IN THIS PROGRAM AND ARE TO BE THE MACHINE
PRECISION EQUIVALENTS OF THE FOLLOWING CONSTANTS.

LDA 1500
LDA 1510
LDA 1520
LDA 1530
LDA 1540
LDA 1550
LDA 1560

```

-1


```

C      E IS A TEST FOR ERRORS OF THE CURRENT ORDER NQ      LDA 2020
C      EUP IS TO TEST FOR INCREASING THE ORDER, EDWN FOR DECREASING THE      LDA 2030
C      ORDER.      LDA 2040
C      LDA 2050
140  K = NQ*(NQ-1)/2
      CALL COPYZ (A(2),COF(K+1),NQ)
      K = NQ+1
      IDOUB = NQ
      ENQ1 = .5D0/NQ
      ENQ2 = .5D0/K
      ENQ3 = .5D0/(NQ+2)
      PEP3H = EPS**2
      E = PERT(NQ,1)*PEPSH
      EUP = PERT(NQ,2)*PEPSH
      ECWN = PERT(NQ,3)*PEPSH
      BND = (EPS*ENQ3)**2
      IWEVAL = 1
      GO TO IRET, (190,200,490,570)
150  IF (H.EQ.HNEW) GO TO 190
      IF CALLER HAS CHANGED H, RESCALE DERIVATIVES TO REFLECT THAT HNEW
      WAS USED ON THE LAST CALL.
      R = H/HNEW
      ASSIGN 190 TO IRET
      GC TO 610
C      SET JSTART TO NQ, THE CURRENT ORDER OF THE METHOD, BEFORE EXIT,
C      AND SAVE THE CURRENT STEP SIZE IN HNEW.
C      LDA 2280
C      LDA 2290
C      LDA 2300
C      LDA 2310
160  JSTART = NQ
      HNEW = H
      RETURN
170  NS = NS+1
      IF (IPRT.LE.0) GO TO 180
      PRINT DATA IF DESIRED BY USER
C      LDA 2370
C      LDA 2380
C      LDA 2390
      PRINT 1, NS,NW,NQ,H,T,(Y(1,I),I=1,NY)
      IF (NL.GT.0) PRINT 2, (YL(I),I=1,NL)
      CONTINUE
180  IF (KFLAG.LT.0) GO TO 160
      IF (T.GE.TEND) GO TO 160
      TAKE ANOTHER STEP IF T < TEND
      JSTART = 1
C      LDA 2450
C      LDA 2460
C      LDA 2470
C      LDA 2490

```

```

C      SAVE DATA FOR TRIAL WITH A SMALLER TIMESTEP IF THIS STEP FAILS      LDA 2500
C      -----LDA 2510
C 190 CALL COPYZ (SAVE,Y,LCOPYY)
C      CALL COPYZ (YLSV,YL,LCOPYL)
C      RACUM = 1.00
C      KFLAG = 1
C      HOLD = H
C      NCOLD = NQ
C      TOLD = T
C      T = T+H
C 200 HINV = 1.00/H
C      -----
C      COMPUTE PREDICTED VALUES BY EFFECTIVELY MULTIPLYING DERIVATIVE
C      VECTOR BY PASCAL TRIANGLE MATRIX
C      -----
C      DO 210 J=2,K
C      J3 = K+J-1
C      DO 210 J1=J,K
C      J2 = J3-J1
C      DO 210 I=1,NY
C      210 Y(J2,I) = Y(J2,I)+Y(J2+1,I)
C      DO 220 I=1,NY
C      220 ER(I) = 0.00
C      -----
C      DO UP TO THREE CORRECTOR ITERATIONS. CONVERGENCE IS OBTAINED WHEN
C      CHANGES ARE LESS THAN BND WHICH IS DEPENDENT ON THE ERROR TEST
C      CONSTANT. THE SUM OF CORRECTIONS IS ACCUMULATED IN ER(I). IT IS
C      EQUAL TO THE K-TH DERIVATIVE OF Y TIMES H**K/(K-FACTORIAL*A(K)),
C      AND THUS IS PROPORTIONAL TO THE ACTUAL ERRORS TO THE LOWEST POWER
C      OF H PRESENT, WHICH IS H**K.
C      -----
C      DO 270 L=1,3
C      CALL DIFFUN (Y,YL,T,HINV,DY)
C      IF (IWEVAL.LT.1) GO TO 230
C      -----
C      IF THERE HAS BEEN A CHANGE OF ORDER OR THERE HAS BEEN TROUBLE
C      WITH CONVERGENCE, PW IS RE-EVALUATED PRIOR TO STARTING THE
C      CORRECTOR ITERATION. IWEVAL IS THEN SET TO -1 AS AN INDICATOR
C      THAT IT HAS BEEN DONE. NEWPW IS SET NONZERO TO INDICATE TO
C      SUBROUTINE NUTSL THAT A NEW PW HAS BEEN PROVIDED.
C      -----

```

```

CALL JACMAT (Y,YL,T,HINV,A(2),N,NY,EPS,DY,F1,PW)
KFLAG = 1
IWEVAL = -1
NW = NW+1
NEWPW = 1
230 CALL NUTSL (PW,DY,F1,N,NY,EPS,YMAX,NEWPW,KRRET)
    IF (KRRET.NE.0) GO TO 600
    IF (NL.LE.0) GO TO 250
C
    DO 240 I=1,NL
240 YL(I) = YL(I)-F1(I+NY)
C
250 CONTINUE
    DEL = 0.D0
C
    DC 260 I=1,NY
    Y(1,I) = Y(1,I)-F1(I)
    Y(2,I) = Y(2,I)+A(2)*F1(I)
    ER(I) = ER(I)+F1(I)
    DEL = DEL+(F1(I)/DMAX1(YMAX(I),DABS(Y(1,I))))**2
260 CONTINUE
C
    IF (L.GE.2) BR = DMAX1(.9D0*BR,DEL/DEL1)
    DEL1 = DEL
    IF (DMIN1(DEL,BR*DEL*2.D0).LE.8ND) GO TO 330
270 CONTINUE
C
    THE CORRECTOR ITERATION FAILED TO CONVERGE IN 3 TRIES. VARIOUS
    POSSIBILITIES ARE CHECKED FOR. IF H IS ALREADY HMIN AND PW HAS
    ALREADY BEEN RE-EVALUATED, A NO CONVERGENCE EXIT IS TAKEN.
    OTHERWISE THE MATRIX PW IS RE-EVALUATED AND/OR (IN THAT ORDER) THE
    STEP IS REDUCED TO TRY AND GET CONVERGENCE.
C
    T = TOLD
    IF (IWEVAL) 280,300,290
    IF (H.LE.HMIN*1.00001D0) GO TO 310
280 RACUM = RACUM*.25D0
290 CONTINUE
300 GO TO 560
310 KFLAG = -3
C
    RESTORE Y AND YL AFTER CONVERGENCE FAILURE
C
320 CALL COPYZ (Y,SAVE,LCOPYZ)
    CALL COPYZ (YL,YLSV,LCOPYL)
    H = HOLD
    NC = NQOLD

```

LDA 3060

LDA 3090

LDA 3120

LDA 3190

LDA 3240
LDA 3250
LDA 3260
LDA 3270
LDA 3280
LDA 3290
LDA 3300
LDA 3310

LDA 3390
LDA 3400
LDA 3410

```

GO TO 170
-----LDA 3470
C THE CORRECTOR CONVERGED, SO NOW THE ERROR TEST IS MADE.-----LDA 3480
C-----LDA 3490
330 D = 0.D0
C
C DO 340 I=1,M1
C YM = DMAX1(DABS(Y(1,I)),YMAX(I))
C 340 D = D+(ER(I)/YM)**2
C
C IWEVAL = 0
C IF (D.GT.E) GO TO 380
C
C THE ERROR TEST IS OKAY, SO THE STEP IS ACCEPTED. IF IDOUB
C NOW BECOMES NEGATIVE, A TEST IS MADE TO SEE IF THE STEP SIZE
C CAN BE INCREASED AT THIS ORDER OR ONE HIGHER OR ONE LOWER.
C THE CHANGE IS MADE ONLY IF THE STEP CAN BE INCREASED BY AT
C LEAST 10%. IDOUB IS SET TO NQ TO PREVENT FURTHER TESTING
C FOR A WHILE. IF NO CHANGE IS MADE, IDOUB IS SET TO 9.
C-----LDA 3580
C-----LDA 3590
C-----LDA 3600
C-----LDA 3610
C-----LDA 3620
C-----LDA 3630
C-----LDA 3640
C-----LDA 3650
C IF (K.LT.3) GO TO 360
C
C DO 350 J=3,K
C
C DO 350 I=1,NY
C 350 Y(J,I) = Y(J,I)+A(J)*ER(I)
C
C KFLAG = 1
C IDOUB = IDOUB-1
C IF (IDOUB) 410,370,510
C 370 CALL COPYZ (ESV,ER,M1)
C GO TO 510
C
C THE ERROR TEST FAILED. IF JSTART = 0, THE DERIVATIVES IN THE
C SAVE ARRAY ARE UPDATED. TESTS ARE THEN MADE TO FIX THE STEP SIZE
C AND PERHAPS REDUCE THE ORDER. AFTER RESTORING AND SCALING THE
C Y VARIABLES, THE STEP IS RETRIED.
C-----LDA 3780
C-----LDA 3790
C-----LDA 3800
C-----LDA 3810
C-----LDA 3820
C-----LDA 3830
C 380 IF (JSTART.GT.0) GO TO 400
C
C DO 390 I=1,NY
C 390 SAVE(2,I) = Y(2,I)
C
C KFLAG = KFLAG-2
C IF (H.LE.HMIN) GO TO 550
C 400
C IF (KFLAG.LE.-5) GO TO 530
C 410 PR2 = (D/E)**ENQ2*1.2D0

```

```

C
L = 0
IF (NQ.LE.1) GO TO 430
D = 0.D0
LDA 3970

C
DO 420 J=1,M1
YM = DMAX1(DABS(Y(1,J)), YMAX(J))
420 D = D+(Y(K,J)/YM)**2
LDA 4010

C
PR1 = (D/EDWN)**ENQ1*1.3D0
IF (PR1.GE.PR2) GO TO 430
PR2 = PR1
L = -1
430 IF (KFLAG.LT.0.OR.NQ.GE.MAXDER) GO TO 450
D = 0
LDA 4080

C
DO 440 J=1,M1
YM = DMAX1(DABS(Y(1,J)), YMAX(J))
440 D = D+((ER(J)-ESV(J))/YM)**2
LDA 4120

C
PR1 = (D/EUP)**ENQ3*1.4D0
IF (PR1.GE.PR2) GO TO 450
PR2 = PR1
L = 1
450 R = 1.D0/DMAX1(PR2,1.D-5)
IF (KFLAG.LT.0.OR.R.GE.1.1D0) GO TO 460
IDUB = 9
GO TO 510
460 NEWQ = NQ+L
K = NEWQ+1
IF (NEWQ.LE.NQ) GO TO 480
R1 = A(NEWQ)/DFLOAT(NEWQ)

C
DO 470 J=1,NY
470 Y(K,J) = ER(J)*R1
LDA 4250

C
480 CONTINUE
LDA 4280

C
IF THE STEP WAS OKAY, SCALE THE Y VARIABLES IN ACCORDANCE
WITH THE NEW VALUE OF H. IF KFLAG < 0, HOWEVER, USE THE
4300 LDA 4310
4310 LDA 4320
4320 LDA 4330
4330 LDA 4340
4340 LDA 4350
4350 LDA 4360
-----
IDUB = NQ
IF (NEWQ.EQ.NQ) GO TO 490
NC = NEWQ
490 ASSIGN 490 TO IRET
GO TO 140

```

```

490 IF (KFLAG.GT.0) GO TO 500
    RACUM = RACUM*R
    GO TO 560
500 R = DMAX1(DMIN1(HMAX/H,R),HMIN/H)
    H = H*R
    IWEAL = 1
    ASSIGN 510 TO IRET
    GO TO 610
C
510 DO 520 I=1,M1
520 YMAX(I) = DMAX1(DABS(Y(1,I)),YMAX(I))
C
    GO TO 170
C
-----LDA 4500
THE ERROR TEST HAS NOW FAILED THREE TIMES, SO THE DERIVATIVES ARE
IN BAD SHAPE. RETURN TO FIRST ORDER METHOD AND TRY AGAIN. OF
COURSE, IF NQ = 1 ALREADY, THEN THERE IS NO HOPE AND WE EXIT WITH
KFLAG = -4.
-----LDA 4530
530 IF (NQ.EQ.1) GO TO 540
    NQ = 1
    ICQUB = 1
    ASSIGN 570 TO IRET
    GO TO 140
540 NCOLD = 1
    KFLAG = -4
    GO TO 320
550 KFLAG = -1
    GO TO 170
C
-----LDA 4550
THIS SECTION RESTORES THE SAVED VALUES OF Y AND YL, SCALING THE
Y DERIVATIVES AS NECESSARY, AND THEN RETURNS TO THE PREDICTOR LOOP
-----LDA 4560
560 H = HOLD*RACUM
570 H = DMAX1(HMIN,DMIN1(H,HMAX))
    RACUM = H/HOLD
    R1 = 1.00
C
    DC 580 J=2,K
    R1 = R1*RACUM
C
    DO 580 I=1,NY
580 Y(J,I) = SAVE(J,I)*R1
C
    DO 590 I=1,NY
590 Y(1,I) = SAVE(1,I)
C
-----LDA 4580
-----LDA 4590
-----LDA 4600
-----LDA 4710
-----LDA 4720
-----LDA 4730
-----LDA 4740
-----LDA 4790
-----LDA 4820
-----LDA 4850
-----LDA 4860
-----LDA 4890

```



```

1 FORMAT (2I5,I2,1P2E10.2,7E14.6/(32X,7E14.6))
2 FORMAT (32X,1P7E14.6)
3 FORMAT (1I,N=,13,1 NL =,13,1 RMSEPS =,1PE9.2,1 TEND =,
4 1E9.2,1 H =,E9.2//) H',8X,'T ',8X,'Y(1,*) AND YL(*)'//)
END
SUBROUTINE COPYZ(S,Y,L)
-----COP
THIS SUBROUTINE COPIES THE ARRAY Y, OF LENGTH L, INTO THE ARRAY S
-----COP
-----COP
-----COP
-----COP

```

C
C
C
C
C

30
40
50
60
70

```

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION S(1),Y(1)
IF(L.LE.0)RETURN
DO 100 J=1,L
  S(J) = Y(J)
100 RETURN
END

```

100

```

SUBROUTINE Derval (Y, YL, T, N, NY, W, KERET)
THIS SUBROUTINE SUPPLIES THE INITIAL VALUES OF THE DERIVATIVES
OF THE NODAL PARAMETERS TAKEN FROM STEADY STATE SYSTEM ANALYSIS.

```

C
C

```

IMPLICIT REAL*8 (A-H,O-Z)
DIMENSION Y(7,1), YL(1), W(1)
Y(2,7)=-9.428681D01
Y(2,8)=-9.079442D01
Y(2,9)=-9.428681D01
Y(2,12)=-1.012731D02
Y(2,13)=-2.894024D02
Y(2,14)=-1.012731D02
Y(2,17)=-1.057558D02
Y(2,18)=-3.616863D01
Y(2,19)=-1.057558D02
Y(2,22)=-7.0622577D01
Y(2,23)=-3.162976D02
Y(2,24)=-7.062277D01
Y(2,27)=-1.322337D02
Y(2,28)=-3.825550D01
Y(2,29)=-1.322337D02
Y(2,42)=0.D0
Y(2,43)=0.D0
Y(2,44)=0.D0
Y(2,47)=0.D0
Y(2,48)=0.D0
Y(2,49)=0.D0
Y(2,52)=0.D0
Y(2,53)=0.D0
Y(2,54)=0.D0

```



```

C C C
RETURN
END
SUBROUTINE DIFFUN(Y, YL, T, HINV, DY)
THIS SUBROUTINE EVALUATES THE SYSTEM'S GOVERNING SET OF
EQUATIONS AT A GIVEN TIME AND GIVEN VALUES OF THE NODAL
PARAMETER AND ITS DERIVATIVE.
IMPLICIT REAL*8(A-H,O-Z,$)
DIMENSION Y(7,1), YL(1), DY(1)
COMMON CD(117,117), TM(117,117), C(117)
DO 200 I=1,117
  DY(I)=-C(I)
DO 100 J=1,117
  DY(I)=DY(I)+CD(I,J)*Y(2,J)*HINV+TM(I,J)*Y(1,J)
CONTINUE
RETURN
C C C
SUBROUTINE JACMAT(Y, YL, T, HINV, A2, N, NY, EPS, DY, F1, PW)
THIS SUBROUTINE EVALUATES THE JACOBIAN MATRIX AT THE GIVEN TIME
AND AT CURRENT VALUES OF THE DEPENDENT VARIABLES, ORDER,
AND STEP SIZE.
IMPLICIT REAL*8(A-H,O-Z,$)
DIMENSION Y(7,1), PW(117,1)
COMMON CD(117,117), TM(117,117), C(117)
AH=-A2*HINV
DO 200 I=1,117
  DO 200 J=1,117
    PW(I,J)=TM(I,J)+AH*CD(I,J)
  RETURN
END
200

```

LIST OF REFERENCES

1. Batchelor, G. K., "Heat Transfer by Free Convection Across a Closed Cavity Between Vertical Boundaries at Different Temperatures," Quarterly of Applied Mathematics, v. 12, p. 209-233, October 1954.
2. Cook, A. H., Interference of Electromagnetic Waves, p. 38-78, Ely House, 1971.
3. DeVahl Davis, G., "Laminar Natural Convection in an Enclosed Rectangular Cavity," Int. J. Heat Mass Transfer, v. 11, p. 1675-1691, 22 April 1968.
4. DeVelis, J. B. and Reynolds, G. O., Theory and Applications of Holography, p. 170, Addison-Wesley, 1967.
5. Eckert, E. R. G. and Carlson, W. O., "Natural Convection in an Air Layer Enclosed Between Two Vertical Plates with Different Temperatures," Int. J. Heat Mass Transfer, v. 2, p. 106-119, 22 January 1960.
6. Huebner, K. H., The Finite Element Method for Engineers, p. 67-118, p. 139-144, p.339-354, Wiley, 1975.
7. Kawahara, M., "Steady and Unsteady Finite Element Analysis of Incompressible Viscous Fluid," Int. J. Numerical Methods in Engineering, v. 10, p. 437-454, 16 January 1975.
8. Merzkirch, W., Flow Visualization, p. 102-184, Academic Press, 1974.
9. Newell, M. E. and Schmidt, F. W., "Heat Transfer by Laminar Natural Convection Within Rectangular Enclosures," J. Heat Transfer, v. 92, no. 1, p. 159-168, February 1970.
10. Oden, J. T., Finite Elements of Nonlinear Continua, p. 118-131, McGraw-Hill, 1972.
11. Schlichting, Hermann, Boundary-Layer Theory, p. 77, McGraw-Hill, 1968.

12. Smith, H. M., Principles of Holography, p. 186-193, Wiley-Interscience, 1969.
13. Steel, W. H., Interferometry, p. 75-95, Cambridge Univeristy Press, 1967.
14. Strang, G. and Fix, G. J., An Analysis of the Finite Element Method, p. 74-95, Prentice-Hall, Inc., 1973.
15. Szekely, J. and Todd, M. R., "Natural Convection in a Rectangular Cavity; Transient Behavior and Two Phase Systems in Laminar Flow," Int. J. Heat Mass Transfer, v. 14, p. 467-481, 18 June 1970.
16. Ural, O., Finite Element Method, Basic Concepts and Applications, p. 164-170, Intext Educational, 1973.
17. Zienkiewicz, O. C., The Finite Element Method in Structural and Continuum Mechanics, p. 148-166, McGraw-Hill, 1967.

INITIAL DISTRIBUTION LIST

| | No. Copies |
|--|------------|
| 1. Defense Documentation Center Cameron Station Alexandria, Virginia 22314 | 2 |
| 2. Library, Code 0212 Naval Postgraduate School Monterey, California 93940 | 2 |
| 3. Department Chairman, Code 67 Department of Aeronautics Naval Postgraduate School Monterey, California 93940 | 1 |
| 4. Professor D. J. Collins, Code 67 Co Department of Aeronautics Naval Postgraduate School Monterey, California 93940 | 2 |
| 5. Dr. H. J. Mueller Naval Air Systems Command AIR 310 Department of the Navy Washington, D. C. 20360 | 1 |
| 6. Dr. Charles M. Vest College of Engineering Department of Mechanical Engineering 550 East University Ann Arbor, Michigan 48109 | 1 |
| 7. LT Gerald Paul Braun, USN 1441 Walbridge Avenue Toledo, Ohio 43609 | 1 |

thesB80189

Time dependent holographic interferometr



3 2768 002 07212 6

DUDLEY KNOX LIBRARY